PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

# INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G07F 7/10, 13/02 | A1

(11) International Publication Number: WO 99/04374

(43) International Publication Date: 28 January 1999 (28.01.99)

(21) International Application Number: PCT/GB98/02083

(22) International Filing Date: 16 July 1998 (16.07.98)

(30) Priority Data:

| | | |
|---|---|---|
| 08/895,225 | 16 July 1997 (16.07.97) | US |
| 08/895,282 | 16 July 1997 (16.07.97) | US |
| 08/895,417 | 16 July 1997 (16.07.97) | US |

(71) Applicant: GILBARCO LIMITED [GB/GB]; Crompton Close, Basildon, Essex SS14 3BA (GB).

(72) Inventor: JOHNSON, William, Smith, Jr.; 911 Fairidge Drive, Jamestown, NC 27282 (US).

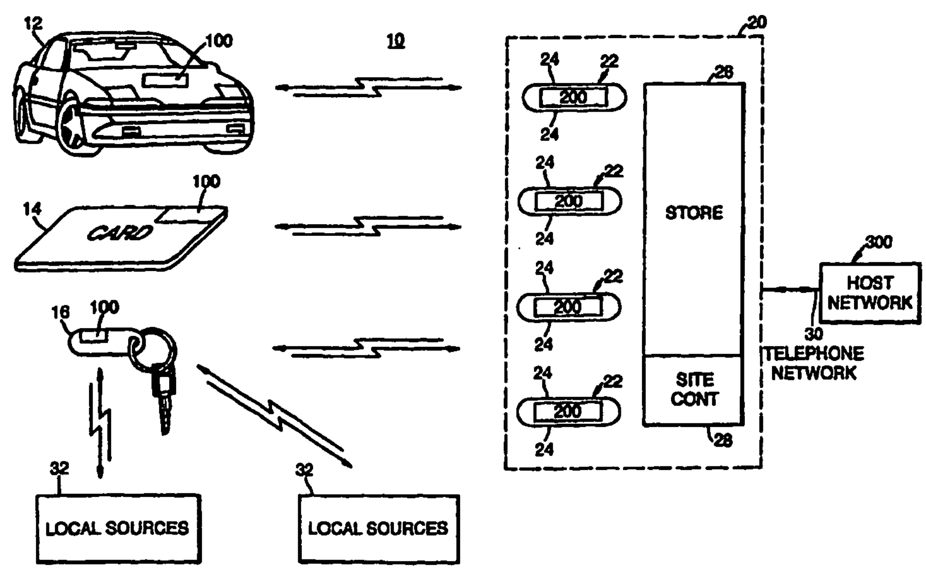
(74) Agent: FITCHETT, Stuart, Paul; GEC Patent Dept., Waterhouse Lane, Chelmsford, Essex CM1 2QX (GB).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published**

*With international search report.*

(54) Title: SECURE TRANSACTIONS




(57) Abstract

The invention provides a secure transaction system including a transponder (100) (tag), point of sale (POS) device (200) and a host network authorization system (300). The transponder and host are arranged to communicate with each other via the point-of-sale device and at least one of the host or transducer authenticates that information received originated from the other, the authentication being based on data available to the host and transponder which data is not available to the point-of-sale device. By employing the invention secure information may be communicated between the transponder and the host via the point-of-sale device without access to the information being possible at the point-of-sale device itself.

BEST AVAILABLE COPY

*FOR THE PURPOSES OF INFORMATION ONLY*

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| AL | Albania | ES | Spain | LS | Lesotho | SI | Slovenia |
| AM | Armenia | FI | Finland | LT | Lithuania | SK | Slovakia |
| AT | Austria | FR | France | LU | Luxembourg | SN | Senegal |
| AU | Australia | GA | Gabon | LV | Latvia | SZ | Swaziland |
| AZ | Azerbaijan | GB | United Kingdom | MC | Monaco | TD | Chad |
| BA | Bosnia and Herzegovina | GE | Georgia | MD | Republic of Moldova | TG | Togo |
| BB | Barbados | GH | Ghana | MG | Madagascar | TJ | Tajikistan |
| BE | Belgium | GN | Guinea | MK | The former Yugoslav Republic of Macedonia | TM | Turkmenistan |
| BF | Burkina Faso | GR | Greece | | | TR | Turkey |
| BG | Bulgaria | HU | Hungary | ML | Mali | TT | Trinidad and Tobago |
| BJ | Benin | IE | Ireland | MN | Mongolia | UA | Ukraine |
| BR | Brazil | IL | Israel | MR | Mauritania | UG | Uganda |
| BY | Belarus | IS | Iceland | MW | Malawi | US | United States of America |
| CA | Canada | IT | Italy | MX | Mexico | UZ | Uzbekistan |
| CF | Central African Republic | JP | Japan | NE | Niger | VN | Viet Nam |
| CG | Congo | KE | Kenya | NL | Netherlands | YU | Yugoslavia |
| CH | Switzerland | KG | Kyrgyzstan | NO | Norway | ZW | Zimbabwe |
| CI | Côte d'Ivoire | KP | Democratic People's Republic of Korea | NZ | New Zealand | | |
| CM | Cameroon | | | PL | Poland | | |
| CN | China | KR | Republic of Korea | PT | Portugal | | |
| CU | Cuba | KZ | Kazakstan | RO | Romania | | |
| CZ | Czech Republic | LC | Saint Lucia | RU | Russian Federation | | |
| DE | Germany | LI | Liechtenstein | SD | Sudan | | |
| DK | Denmark | LK | Sri Lanka | SE | Sweden | | |
| EE | Estonia | LR | Liberia | SG | Singapore | | |

# SECURE TRANSACTIONS

The present invention relates generally to providing secure transactions between a host, a local transaction device and a remote communication device and, more particularly, to a transponder fuel dispensing system for providing secure authorizations and transactions using the transponder in a fuel delivery, retail sales and service environment.

In recent years, traditional gasoline pumps at service stations have evolved into elaborate point-of-sale (POS) devices having sophisticated control electronics and user interfaces with large displays and touch pads (or screens). These dispensers include various types of payment means, such as card readers, to expedite and further enhance fueling transactions. A customer is not limited to the purchase of fuel at the dispenser. More recent dispensers allow the customer to purchase services, such as car washes, and goods such as fast food or convenience store products at the dispenser. Once purchased, the customer need only pick up the goods and services at the station store.

Given the ever increasing demand to increase transaction efficiency by both fuel suppliers and customers, transaction systems associated with the service stations are further evolving to provide fully automated authorization and purchasing. It would be advantageous if customers no longer needed to use a credit/debit card or smartcard to purchase fuel or other products or services. This can be accomplished if the customer, vehicle or both are equipped with a remote intelligent communications device, or transponder (hereinafter referred to as a tag for simplicity), capable of remotely communicating with fuel dispensers and other devices as desired. These tags and dispensers operate in conjunction to provide a

cashless and cardless transaction system where transactions are automatically charged or debited without requiring any action by the customer.

Numerous applications and patents relate to technology associated with communicating information between a tag or like transponder and the fuel dispenser. These patents disclose communicating between the tag and fuel dispenser with fiber optics, electromagnetic radiation, such as radio frequency transmissions, infrared, direct electrical connections and various others means or combination of these means. Various types of information is communicated between the tag and the dispenser including vehicle identification, customer identification, account information, fuel requirements, diagnostics, advertising, and various other types of solicited and unsolicited messages. Certain specific applications equip the tag and dispenser with cryptography electronics to encrypt and decrypt data transferred between the tag and dispenser.

Tag transponder technology is used in many areas of technology relating to vehicles. Such technology is used in tracking vehicles, navigational aids, toll collection, diagnostics, vehicle security and theft deterrence, keyless entry, refueling, collision avoidance, vehicle identification, surveillance and traffic control as well as transmitting and receiving financial data.

Using tag technology in association with the fuel dispensing environment is currently in its embryonic stage. Although further advances occur at a continually increasing rate, a concern of utmost importance has not been addressed. Given the ingenuity and resourcefulness of information thieves, financial and account information is susceptible to theft if transmitted from the tag to the dispenser and onto a host network system, which generally provides authorization and account adjustment functions for the transaction. Even with today's most sophisticated coding and encryption techniques, information thieves,

given enough time and opportunity, are able to decrypt the encrypted data and obtain various financial and account information.

Further dangers arise when each fuel dispenser either includes, or relays this account information to the host network. Theft and fraud opportunities arise from not only the information thieves stealing information from the airwaves in the local transaction environment, but from those having access to confidential information stored in the tag, dispenser and fuel station store. A stolen tag provides a thief an indefinite amount of time to figure out the internal workings and information within the tag.

Additionally, since most of these tags will necessarily and ultimately communicate with a central network host through the fuel dispenser POS, access to various different accounts and different financial information is available for purchasing any number of products using a number of accounts. In addition to providing high levels of security for tag authorization, account access, and data transfer, less sophisticated security techniques providing lower levels of security are needed to facilitate certain tag functions that either do not require a high degree of security or require the tag to access and communicate with a fuel dispenser or other source, such as a restaurant or other goods or services provider that does not have high level security capability. For example, service stations or restaurants may want to access various types of non-confidential information on the tag to determine identification or other information relating to their particular business. Based on this information, the businesses may want to provide loyalty points for a tag holder relating to the number of visits or amount of goods or services purchased. Although providing loyalty points does not require the intense security necessary to secure financial account information, these local businesses still need to ensure to a sufficient degree that unscrupulous customers cannot adjust their loyalty points at will.

Another reason for providing different levels or types of security for different transactions is to reduce the number of times information thieves have access to a given security technique. Security is enhanced if the manner of encryption or the key used for encryption varies periodically and if the key used for encryption and decryption is never transmitted. Thus, there is a need for a secure transaction system capable of providing virtually impenetrable transactions between a tag and a fuel dispenser associated with a host authorization network. There is also a need for this system and tag to enable other sources to access the tag at lower levels of security without jeopardizing account or financial information in addition to providing general access to non-confidential information. The system should avoid transmitting encryption keys and should periodically change keys during transactions to minimize any chance a thief may have in deciphering the key.

According to the present invention there is provided a system for providing secure transactions between a host, a point-of-sale device and a remote transponder, wherein the transponder and host are arranged to communicate information and/or instructions between each other via the point-of-sale device, characterised in that at least one of the host or transducer authenticates that the information and/or instruction received originated from the other, the authentication being based on data available to the host and transponder which data is not available to the point-of-sale device. The authentication may be by way of the data being encrypted or associated with a passsword.

Preferably the transponder, (tag) is adapted to bi-directionally communicate with a POS device, preferably a fuel dispenser, which further communicates with a host network to provide authorization of the tag and carry out any desired purchases or transactions. To

avoid transmitting data from which valuable account or financial information could be derived between the tag and POS device or the POS device and the host network system, the invention may maintain all or a majority of account and financial information requiring absolute security only at the host network. In the preferred embodiment, neither the tag nor the POS device has, or has access to, critical financial or account information. But, the security system will also provide high levels of security for applications requiring transmission of such information.

In order to avoid placing this information at risk during transactions, the invention preferably provides a unique identifier for each transponder wherein the host network maintains account and financial information associated with the tag having the unique identifier. The tag identifier is transmitted to the host system through the POS device, and the host network checks to see that the tag, and not a counterfeit, has sent the identifier. Once the host system determines that an authorized tag sent the identifier, the host authorizes the POS device to further interact with the tag and allow all or certain subsequent transactions.

Preferably, the tag is authenticated using identical cryptography techniques known only by the tag and host, but not by the POS device. Initially, the communication electronics of POS device, acting as an interrogator, will continuously scan for a tag within the field. Once a tag comes within the field in response to the interrogator, the interrogator will recover the tag's identifier, hereinafter referred to as the ID number, from the tag. The POS device will generate authentication check data, preferably a random number, and send it to the tag for encryption. The tag then encrypts the random number with an encryption technique using a main encryption key and transmits the encrypted random number back to the POS device. The interrogator passes the ID number, the encrypted random number and the original

random number to the host through the associated POS device. The host determines or calculates the main encryption key used in the tag from the tag ID number. The host encrypts the random number sent from the POS device and compares it with the encrypted random number sent from the tag through the POS device. If these numbers match, the host signals the POS device that the tag is valid.

To further enhance security, the main encryption key stored in the tag and determined or calculated by the host is only used for authentication of the tag. Data transfers or transactions between the tag and host through the POS device requiring security are encrypted and decrypted as necessary by the tag and host using a session key, which is different then the main encryption key. Preferably, a new session key is generated for each transaction and is a function of a random number, independently generated in the tag and host, as well as the main encryption key. The tag random number is preferably generated upon receipt of the random number generated at the POS device. The tag random number is transmitted to the host through the POS device so that both the tag and host can independently generate the session key capable of encrypting and decrypting data at either the host or the tag. The method used to generate the session key from the tag random number must be the same in both the host and tag. Similarly, the electronics in the tag and host are used to encrypt the random number generated at the POS device.

Each tag includes memory for storing various types of data, including information and commands. Preferably the memory and associated electronics allow data to be stored in the memory, read from the memory and transmitted to the POS device and altered according to commands sent from remote sources, including, but not limited to the POS device.

For example, the tag may include portions of memory which are accessible and modifiable by numerous fuel dispensers, restaurant interrogators and the like. Various portions of

memory may have different security levels with corresponding passwords. Preferably, the memory is partitioned into four or more groups wherein the associated electronics may store data in the group, read data from the group, add to the data in the group and subtract from the data in the group. Depending on the group, the available functions are controlled by the group and its security level. For example, a first group (group I) may require all functions to be secure; thereby, requiring that only the host, acting through the POS device, has access to the group I partition and that only the host can carry out the store, read, add or subtract functions. A second group (group II) may require the store, add and subtract functions be authorized by the host and allow sources other than the host to read the data stored in group II. This is useful with tags used as smartcards having a cash reserve or prepaid tags having a set amount of credit. Group II may be arranged so that sources other then the host can check availability of funds, but cannot modify those funds without authorization. A third group (group III) may be configured so that the store and add functions are secure while the read and subtract functions are accessible by other sources at a lower security level. This lower security may be provided with cryptography and/or passwords. Group III data arrangements are useful where sources other than the host recognize customer loyalty and are allowed to provide the customer benefits based on a number of loyalty points accumulated on the tag. The source may subtract a loyalty point for a benefit provided, as well as ensure the benefit should be provided by checking the loyalty points stored on the tag. However, it is undesirable to allow the local source to add loyalty points without authorization. A fourth group (group IV) may be arranged such that customer information is accessible by sources other than the host, but cannot be changed without authorization from the host. Group IV is useful in situations where local sources need to determine information, such as identification, about the tag and tag holder.

These and other aspects of the present invention will become apparent to those skilled in the art after reading the following description of the preferred embodiments in conjunction with the drawings.

One embodiment of the present invention will now be described, by way of example only, with reference to the accompanying figures, in which like numerals are used throughout, and of which:

FIGURE 1 is a schematic of a tag constructed and implemented according to a preferred embodiment of the present invention interacting with a POS device and host network.

FIGURE 2A is a schematic representation of the tag 100 constructed according to the preferred embodiment.

FIGURE 2B is a schematic of a POS device and host network constructed according to the preferred embodiment.

FIGURE 2C is a schematic representation of the tag 100 having integrated electronics constructed according to the preferred embodiment.

FIGURE 3A shows a preferred format for a tag ID.

FIGURE 3B shows sample tag ID values for bytes 5 and 6 for the tag ID format shown in Figure 3A.

FIGURES 4A TO 19 illustrate various processes and data organisation employed in a preferred embodiment of the present invention.

Referring to the drawings, as best seen in Figure 1, a secure transaction system generally designated 10 includes or is associated with three major subsystems: a remote

communication unit 100 (hereinafter a tag); a POS device 200 and a host network 300. In general, remote communication units 100 are adapted to communicate with and through the POS device 200 in order to obtain authorization and communicate information to and from the host network 300.

Various means of security are employed depending on the information being communicated and the source and destination of the information. Importantly, the tag 100 and host network 300 are adapted to encrypt and decrypt certain communications there-between, while the POS device 200 primarily only relays the encrypted information sent between the tag 100 and host network 300. The POS device 200 is unable to decrypt such information.

The tag unit 100 is integrated into a small carrying medium, such as a module in a vehicle 12, a transaction card 14 or a key fob 16. The tag 100 provides remote bi-directional communications with the POS device 200. POS devices 200 are located at each fueling position 24 associated with fuel dispenser 22 of fuel dispensing environment 20. The dispensers are operatively associated with a central station store 26 by a conventional wire system, which may house a convenience store as well as one or more restaurants. Many fuel dispensing environments 20 will provide services, such as car washes, in addition to goods. Generally the store 26 will include a central site controller 28 to provide central control functions for each dispenser 22. Each dispenser, and its respective POS electronics, generally communicates either directly, or indirectly through the central site controller 28, to the host network 300 via a telephone network 30. The host network 300 provides authorizations and other data for the various transactions attempted at each POS device 200. In addition to communicating with the POS devices 200, the transponders 100 also communicate with various other local sources 32 for various informational and transaction-

type functions. These local sources 32 may include any number of goods or service providers, such as local restaurants.

The present invention provides virtually impenetrable security with respect to informational and financial communications between the transponder 100 and the host network 300 through the POS device 200. High levels of security are also provided as desired for communications solely directed to the POS device 200 and the local sources 32 from the tag 100. Such direct communications from the tag 100 to the POS device 200 are those communications not intended to require interaction with the host network 300.

Turning now to Figure 2A, the tag 100 is shown. Communications electronics 102, provide remote communications with various remote sources, includes a transmitter 106 and receiver 108 having associated antennas 110, 112. The transmitter 106 and receiver 108 operate to transmit and receive data to and from the remote communications unit 100. The communications electronics 102 also include a battery power supply 114, a communication controller 116 associated with a memory 120 having the software 122 necessary to operate the communications electronics 102 and communicate with the cryptography electronics 104. Serial communications between the communication electronics and the cryptography electronics 102, 104 is provided via the input/output (I/O) ports 124, 138 associated with the respective electronics. The communication electronics 102 provide a clock 128 signal to the I/O port 138 of the cryptography electronics 104. The cryptography electronics 104 include a controller 130, memory 132 and software 134 necessary to encrypt and decrypt data, as well as provide any additional operations. The memory 120, 132 may include random access memory, read only memory or a combination of both. Notably, the communication controller 116 and the cryptography controller 130 may be integrated into one controller.

Similarly the software and memory of the communication and cryptography modules could be merged.

As shown in Figure 2C, the communication and cryptography electronics, as well as any associated controllers may be integrated into a single controller system and/or integrated circuit. In such cases, a single controller 115 is associated with memory 117 having software 119 as necessary for operation. In such an integrated system, the controller 115 will carry out any cryptography functions as well as any other necessary functions.

The communication electronics 102 are preferably the Micron MicroStamp™ produced by Micron Communications, Inc., 8000 South Federal Way, Boise, Idaho 83707-0006. A detailed description of the MicroStamp engine™ is provided in the Preliminary Data Sheet and the MicroStamp Standard Programmers Reference Manual provided by Micron Communications, Inc.

In order to save power and extend battery life, the communication electronics 102 operate at a low-current, sleep mode until an internal programmable timer causes it to wake up. The communication electronics 102 determine whether there is a properly modulated signal present and, if not, immediately returns to the sleep mode. The modulated signal of which the communication electronics 102 look for once it awakens is provided by the POS device 200 or one of the local sources 32. If a properly modulated signal is present, the communication electronics 102 process the received command and send an appropriate reply. The communication electronics 102 then return to the sleep mode. The communications electronics 102 cause the cryptography electronics 104 to awaken as necessary to encrypt or decrypt data received by or transmitted from the tag 100.

Referring now to Figure 2B, a schematic representation of the POS device 200 and host network 300 is shown. The POS device 200 preferably includes a controller 202 forming

communication electronics 204 and cryptography electronics 206. The controller 202 is associated with a memory 210 and an antenna 208 for providing remote communications. The controller 202 interfaces with the telephone network 30 to provide bi-directional communications with the host network 300. The POS device 200 includes a display 212 and an input device 214, such as a touch pad or touch screen associated with the display 212. The POS device 200 is a fuel dispenser having at least two fueling positions and a card reader 216 for receiving payment through any variety of credit, debit or smartcards and a cash acceptor for receiving payment. The card and cash acceptors are present to service those customers who do not have tags 100.

The host network 300 includes a control system 302 forming communication electronics 304 and cryptography electronics 306. The host network 300 also includes memory 310 associated with the electronics 304, 306. The host network 300 may include additional communications ports 312 for communicating with other POS devices.

**TAG OVERVIEW**

Preferably, the tag cryptography electronics 104 implement the encryption standard called DES (or Data Encryption Standard) as defined by US Federal Information Processing Standard Publication Number 46. Various commands are available in the tag cryptography electronics 104 using the DES algorithm as a basis for security. Other cryptography methods are available and will work with the current invention.

The tag cryptography electronics 104 are used for two main functions in the tag 100. First, it provides a secure authentication procedure when implemented with the network host

300. Thus, if the tag 100 passes this authentication procedure, then to a high level of confidence, the host 300 is assured that a real tag 100 was authenticated. Second, the tag cryptography electronics 104 allow the host 300 to securely write data to the tag 100, read data from the tag 100 or modify data in the tag 100.

Attention is directed again to Figure 2A. When DES authentication and write functions are included in the tag, all of the DES functionality, including storage of DES keys, resides in the tag cryptography electronics 104 Communications from the communication controller 116 to the tag cryptography electronics 104 takes place via the tag communication electronics 102 using the WriteDigitalPort and ReadDigitalPort commands, which are standard MicroStamp commands.

All data, including commands sent to the tag cryptography electronics 104, are sent as data within the WriteDigitalPort command. The data to be sent to the tag cryptography electronics 104 is stored in a buffer in the tag communication electronics 102 until it is serially transmitted to the tag cryptography electronics 104. The data packet sent to the tag cryptography electronics 104 using the WriteDigitalPort command is only limited by the size of the buffer in the tag communication electronics 102 and is preferably 64 bytes.

Likewise, when the tag cryptography electronics 104 have data for the tag communication electronics 102, data is transmitted to the tag communication electronics 102 where it is stored in a buffer in the tag communication electronics 102. The size of the data packet from tag cryptography electronics 104 are limited in the same manner set forth above with regard to the data packet sent to the tag cryptography electronics 104. This data is read from the MicroStamp module using the ReadDigitalPort command.

## HOST TAG GENERATION

When initially configuring tag 100, the host network 300 generates a tag identification number (ID) and a main tag key, which is preferably a DES key, and directly injects these numbers into the tag 100 along with any other pertinent information. Generally, the POS device 200 is not used during configuration. Preferably, a different main tag key is injected into every tag 100, and a secure algorithm is used to generate the main tag keys from a master key known only by the host 300. Maximum security is obtained when it is impossible for an infiltrator or cracker to calculate the host master key from the main tag key and tag ID.

The preferred format for a 10 byte tag ID is shown in Figures 3A and 3B. The first four bytes, bytes 1-4, are reserved for market segment definitions. Preferably, each market segment includes a classification as defined by the *Standard Industrial Classification Manual*. For example, the classification (generally known as the SIC code) for the service station industry is "5541". This classification is preferably stored as four byte ASCII value (hex 35 35 34 31). In this example, the code would appear in the first four bytes of all tags of companies in the service station industry. Bytes 5 and 6 identify the issuing company and the last four bytes are represent the specific tag's generation number used to uniquely identify the particular tag for that company. With bytes 7-10 available to define a tag number for a particular company, over four billion tags may issue without repetition for each issuing company. Special codes may be stored in any of the tag ID fields, such as the issuing company field, to provide for local or network testing. If the tag ID is formatted as depicted and the fifth and sixth data fields are coded 00 (hex 30 30), then the tag is a local test tag, and therefore, the tag ID is not passed on to the host 300. If the fifth and sixth data

fields are coded 01 (hex 30 31), then the tag is a network test tag and the tag ID is passed on to the host 300. Such testing is typically used to test communication ability and assure proper operation of the POS device 200 and/or host 300.

## PREFERRED DES KEY GENERATION PROCEDURE

The preferred procedure for generating each main tag key for the various tags 100 employs triple encryption with three secret master keys. The main tag keys are generated by the host 300. Referring now to Figure 4A, the last 8 bytes of the tag ID number are preferably used to generate the main tag key. For example, if Shell Oil Company is the issuing company, the 8 byte number (rendered here in hexadecimal) would be of the form 34 31 53 48 xx xx xx xx, where the x's represent the individual tag's sequence number. Initially, an exclusive-OR (XOR) is taken of the 8 byte number derived from the tag ID and a constant, chosen randomly, whose value is securely held by the host network 300. The result is XORed with the first master key (Mkey #1), **encrypted** with the first master key, and XORed with the first master key. This result is XORed with the second master key (Mkey #2), **decrypted** with the second master key, and XORed with the second master key. This result is XORed with the third master key (Mkey #3), **encrypted** with the third master key, and XORed with the third master key. The final result is the main tag key, which is loaded into the tag 100.

All four components of the key generation procedure, the constant, first master key (Mkey #1), second master key (Mkey #2), and third master key (Mkey #3) are held in strict confidence at the host 300. These components are not cryptographically or mathematically

related to each other. Furthermore, the host 300 should ensure that none of these components are cryptographically decipherable by unauthorized persons. Although this is the preferred embodiment, various combinations of logic functions and cryptography are possible to ensure a difficult to crack key generation process. As with any of the key and code generation techniques disclosed herein, portions of the tag ID may be operated on by a function to achieve a desired result. The function may range from f(x)=x to any convoluted function chosen for the generation process.

The preferred process for loading the tag 100 with its respective tag ID and main DES key is shown in Figure 4B. As discussed above, a 10 byte tag ID number comprising the standard industry code, company code and individual sequence number is directly loaded by the host 300 into the tag 100 during generation of the tag. Typically, a 4 byte sequence counter generates the tag sequence number forming bytes 7-10 of the tag ID. Concurrently with loading the tag ID number, the main DES tag key is generated using the master keys and loaded into the tag 100. Additionally, other information is loaded into the tag 100 during loading as desired. Once the tag is generated, it is distributed to customers for use with the POS device 200 and local sources 32.

## HOST AUTHENTICATION OF TAG

Typically, tags 100 are authenticated at the host 300 using a DES encrypted value produced by the tag 100. The main tag key for this DES calculation is stored in the tag cryptography electronics 104 when the tag 100 is initialized. As noted, the tag 100 is

initialized directly at host 300 without transmitting the main tag key to or through any other system.

A preferred tag authorization process is shown in Figure 5. In step 1, the POS communication electronics 204 of the POS device 200 generate and send a random number (CRN) to the tag 100 and to the host 300. The tag 100 encrypts the random number and returns the encrypted random number (TRN) to the POS device 200 along with a tag identification number (ID) in step 2. The POS communication electronics relays the tag ID, the encrypted random number received from the tag 100 and the random number to the host 300 without modification in step 3. In the preferred embodiment, the tag ID number is 10 bytes, the random number is 8 bytes and the encrypted random number is 8 bytes. Upon receipt of the tag ID from the POS device 200, the host 300 calculates (or looks up) the main tag key for the tag 100 using the tag ID and the secret master keys in the same manner as the main tag key was initially created from the tag ID. In other words, the host 300 recalculates the main tag key for each operation. The host cryptography electronics 206 encrypt the random number using the recalculated main tag key and compares the result to the encrypted random number received from the tag 100. If they match, the tag 100 is a valid tag, and most likely not a counterfeit. Upon authorization, the host 300 can use the ID number to look up transaction billing data or other information associated with the tag 100 or its owner and authorize the POS device 200 to carry the desired transactions in step 4.

In a fueling environment, the POS device 200 is, or is incorporated within, a fuel dispenser 24. Typically the card reader 216 in the dispenser (CRIND) will operate in conjunction with the POS communication electronics 204 to provide an interrogator. In order for the host 300 to authenticate a tag 100, the POS communication electronics 204 will

continuously scan for a tag 100 within a particular communications field or range. The tag 100 will respond once inside the field by sending a command to POS device 200. This interrogation and response sequence is conventional in the Micron MicroStamp System.

Once the tag 100 responds, the POS device 200 will recover the tag ID from the tag 100. The POS device 200 generates the random number (CRN) and sends it to the tag 100 with an appropriate command. The tag 100 then encrypts the random number (CRN) using DES with the main tag key and transmits the encrypted random number (ECRN) back to the POS device 200. The POS device 200 subsequently transmits the tag ID, the random number (CRN), and the encrypted random number (ECRN) to the host 300.

The host 300 calculates the main tag key from the ID number in the same manner in which the host 300 originally generated the main tag key (see Figure 4). The random number (CRN) originally generated by the POS device 200 is encrypted by the host 300 to provide a host network encrypted random number (NERN). The host 300 then compares the encrypted random number (ECRN) encrypted at the tag 100 to the host network encrypted random number (NERN). If the ECRN and NERN match, then the host signals the POS device 200 or site controller 28 that the tag 100 is valid and authorized. Alternatively, the host 300 may decrypt the encrypted random number (ECRN) and compare the result to the random number (CRN).

## DES MODULE REGISTERS

The tag cryptography electronics 104 and memory 132 of the tag 100 includes several designated registers for storing various types of values and information. Certain of these registers used in the preferred embodiment are listed and described as follows:

UNSIGNED CHAR DES Key Counter;

UNSIGNED CHAR Current Sequence Number;

UNSIGNED CHAR DES Password[8];

UNSIGNED CHAR DES Key[8];

UNSIGNED SHORT DES Module user Memory Size;

UNSIGNED CHAR DES Module Version Number;

UNSIGNED SHORT BEGIN BLOCK GROUP 0;

UNSIGNED SHORT END BLOCK GROUP 0;

UNSIGNED CHAR GROUP MODE 0;

UNSIGNED SHORT BEGIN BLOCK GROUP 1;

UNSIGNED SHORT END BLOCK GROUP 1;

UNSIGNED CHAR GROUP MODE 1;

UNSIGNED SHORT BEGIN BLOCK GROUP 2;

UNSIGNED SHORT END BLOCK GROUP 2;

UNSIGNED CHAR GROUP MODE 2;

UNSIGNED SHORT BEGIN BLOCK GROUP 3;

UNSIGNED SHORT END BLOCK GROUP 3; and

UNSIGNED CHAR GROUP MODE 3.

- **DES Key Counter**

The DES key counter is one byte in size and is incremented only when a new main tag key (a DES key) is written to the tag cryptography electronics 104. The DES key counter is initially zero when the tag cryptography electronics 104 is first powered up. This register allows an authorized tag programmer (i.e. host 300) to calculate a current DES password and main tag key and to change the main tag key if necessary. The DES password and its function is described in greater detail below.

- **Current Sequence Number**

The current sequence number is used when transferring data between the tag 100 and a local interrogator of the POS device 200. The current sequence number allows the interrogator to ensure that an event has occurred in the presence of noise or other disturbing factors and prevents a command or operation from being accidentally replicated. The current sequence number preferably corresponds to a tally of transactions or operations the tag has conducted. The host typically sends a sequence number to the tag with each command. If the sequence number sent by the host does not correspond to that stored in this register, the command is ignored. Typically, the host will include a new sequence number, one plus the current sequence number, to signify a new command affecting store data is being sent. Read commands preferably include the current sequence number. The register

in the tag is updated accordingly. The sequence number adds further security by ensuring the host 300 is transmitting proper data and/or commands to the tag, that no unauthorized communications have occurred and that each previous communication between the host and tag was properly received and acted on.

- **DES Password**

The password prevents unauthorized replacement of the main tag key in the DES Module. The password is used to gain access to the key, while the key is what is used by the cryptography electronics to encrypt information. If an attacker could replace the main tag key, various registers and memory locations could be altered to allow the attacker to enjoy secondary services such as loyalty points and other benefits from sources other than the host without having earned them. A loyalty plan may be a program where customers collect bonus points based on purchases and transactions. The bonus points may be stored on the tag and redeemed as desired for benefits or privileges at the fuel station store or any other local source 32. Notably, even if an attacker could replace the main tag key, an authentication check with the host 300 would fail since the host 300 would generate the original main tag key or one authorized and updated by the host 300. For authorization and other secure transactions, the host 300 and tag 100 must have or be able to generate identical keys. Since some loyalty plans require a non-secure read function, the DES password may be useful to access read-only information stored on the tag 100.

Preferably, the DES password register can only be directly written once. After that, the tag cryptography electronics 104 refuse further attempts to write to the DES password register. The DES password is modified by the tag 100 when a main tag key is initially

written to the tag 100. The tag 100 preferably uses the new main tag key to encrypt the previous DES password and write the encrypted result into the DES password register to form a new password.

**- Main Tag Key**

As noted, the main tag key is preferably an 8 byte standard DES encryption key that is used by the tag 100 for both encryption and decryption. The main tag key is loaded in the tag 100 with a set DES key command as discussed in the "DES commands section" below.

**- User Memory Size**

The memory size register stores a value representative of the amount of memory available in the tag cryptography electronics 104. The memory size is preferably reported in blocks, with each block having 8 bytes of data.

**- DES Version Number**

The version number register is a read only register and stores the cryptography electronics 104 software and/or hardware version.

**- Begin and End Block Registers**

The begin and end block registers are generally set by the tag programmer to reflect desired groupings of the user memory in the memory 120 of the tag cryptography electronics

104. These registers preferably reflect the begin and end blocks of the groups starting from memory location zero. A group is unused or does not exist if the begin and end block registers for that group are both set to zero. Preferably, there are four groups and one set of Begin and End registers for each group.

- Group Mode Register

The group mode register defines the acceptable commands for a particular group. There is a group mode register for each of the groups. Preferably, there are 8 total commands selectable by this register. These commands and their respective bit positions are listed in Figure 6. The group mode register is an 8 bit register. Each bit position in the register refers to one specific command. A logical '1' in a given bit position indicates that the respective command is selected for the respective group. For example, the group mode register depicted in Figure 7 is programmed to 0F hexadecimal (HEX). This means that the group will accept all four of the secure commands (read, subtract, add, and write) but none of the unsecure commands.

## DES TAG COMMANDS

The tag 100 may receive and respond to various cryptography related commands, such as those shown in Figure 8. The DES commands in bold preferably require cryptography, either encryption or decryption, to encode or decode data.

As depicted in Figure 9A, when the tag 100 is presented with a random number (CRN) from the POS device 200 during an authentication cycle, the tag 100 also generates a tag random number (TRN) that is sent to the host 300 through the POS device 200. This tag random number (TRN) is used in the tag 100 to generate a session key (SK) from the main tag key of the tag 100. The host 300 uses its knowledge of the tag's original main tag key to generate the same session key (SK) using the tag random number (TRN). The session key (SK) is used to encrypt or decrypt actual data and authenticator data transferred between the tag 100 and the host 300. Actual data represents the actual information or commands to be transferred. Authenticator data is the result of a security or checking process performed on the actual data to allow the receiving device, whether the host 300 or tag 100, to assure accuracy of data transmission. The process is described in detail below.

The session key (SK) is preferably used a maximum of four times before it is destroyed in the tag 100. After that, a new tag random number (TRN) is generated by performing another tag authentication cycle using a new random number (CRN) generated by the POS device 200. Requesting a tag authentication cycle produces a new tag random number (TRN) regardless of how many times it has been used. Note that the tag authentication cycle or process uses the encrypt random number command, the third command listed in Figure 8, to encrypt the random number (CRN) received from the POS device 200.

For secure commands transferring data from the host 300 to the tag 100, through the POS device 200, data is encrypted at the host with the session key (SK) generated during the tag authentication process and decrypted at the tag 100 using an identically calculated session key (SK). As noted, the session key (SK) is generated at the host 300 by encrypting

the tag random number (TRN) with the main tag key. Session key (SK) generation may include other logic functions in addition to encryption. In this context, encryption is meant to include any additional functions used to arrive at the session key (SK) or like value. Data to be authenticated is generally 8 bytes long and is assumed to be numeric but generally no checks are performed to ensure that it is.

The authenticator data is preferably formed by concatenating a 4 byte, 32 bit cyclic redundancy check (a super checksum process referred to as CRC) on the original data, with 4 bytes of random numbers generated by the host 300. The CRC occupies the least significant 4 bytes of the 8 byte authenticator data. The authenticator data should also be encrypted by the host 300 using the tag's session key (SK). When the tag 100 decrypts the authenticator data and checks the accuracy of the transfer, the 4 bytes of random numbers generated by the host are discarded and only the CRC bytes are used in checking the data.

For secure commands transferring data from the tag 100 to the host 300, through the POS device 200, as shown in Figure 9B, data is encrypted with the session key (SK) generated during the tag authentication process. Again, the session key (SK) is generated by encrypting the tag random number (TRN) with the main tag key. The data to be authenticated is generally 8 bytes long and assumed to be numeric. As above, authenticator data is used to check the accuracy of the data transfer. The authenticator data is formed by concatenating a 4 byte, 32 bit CRC on the data, with 4 bytes of random numbers generated by the tag 100. The CRC occupies the least significant 4 bytes of the 8 byte authenticator data. The authenticator data is encrypted by the tag 100 using the tag's session key (SK). When the host 300 decrypts the authenticator data, the 4 bytes of random numbers generated by the tag 100 are discarded and only the CRC bytes are used in checking the data.

Additional error checking and detection for either tag-to-host or host-to-tag communications is provided by using a longitudinal redundancy check (LRC). An LRC appended to each communications block is formed by performing a byte-wise binary addition of every preceding byte of data in the communications block. Carries from the additions are discarded.

**- Set DES Password**

The set DES password command allows a tag programmer to load the DES or other password in the tag 100. The tag 100 will not accept a main tag key if the password is zero. The password is set to zero at initialization. The password can only be set once. After it is set, new passwords are generated by the tag. The following command protocol sequence is recommended from a tag programmer associated with the host 300 to initially set the DES password:

| | |
|---|---|
| UNSIGNED CHAR Set_DES_Password = E1 | (command to set password); |
| UNSIGNED CHAR DES_Password[8] | (actual 8 byte password); and |
| UNSIGNED CHAR LRC | (checksum). |

Possible replies from the tag 100 are:

UNSIGNED CHAR Command_Accepted = ACK; and

UNSIGNED CHAR Command_Not_Accepted = NAK.

- **Set DES Key**

The set DES key command allows the tag programmer to directly set the main tag key for use in all subsequent secure communications with the tag 100. Preferably, the DES password is sent as part of this command sequence. The command protocol sequence follows:

| | |
|---|---|
| UNSIGNED CHAR Set_DES_Key = E2 | (command to set main tag key); |
| UNSIGNED CHAR DES_Password[8] | (send actual password); |
| UNSIGNED CHAR DES_Key[8] | (send main tag key); and |
| UNSIGNED CHAR LRC | (checksum). |

Possible replies from the tag 100 are:

UNSIGNED CHAR Command_Accepted = ACK; and

UNSIGNED CHAR Command_Not_Accepted = NAK.

When the tag cryptography electronics 104 of the tag 100 accept a main tag key, the main tag key is used to encrypt the current DES password. The result of that encryption is then stored as the new DES password. In addition, each time a DES key is accepted, the DES key counter is incremented by one. The DES key counter is zeroed at initial power on

for the tag, and cannot be directly written or modified except by writing a new DES key to the tag cryptography electronics 104.

**- Set Group Registers**

The set group registers command allows the tag programmer to set up group information for the user memory in the DES Module. The DES password is given as part of this command. This command may be configured to work even if the DES password is zero (i.e. before the DES password is loaded into the tag cryptography electronics 104). Note that the block numbers defined in this command are absolute, start from block zero (0) and do not relate to the group number. The begin and end block numbers may overlap.

The following is a sample command sequence for setting four group registers:

| | |
|---|---|
| UNSIGNED CHAR Set_DES_Key = E3 | (set main tag key); |
| UNSIGNED CHAR DES_Password[8] | (send password); |
| UNSIGNED SHORT BEGIN_BLOCK_GROUP_0 | (start first block); |
| UNSIGNED SHORT END_BLOCK_GROUP_0 | (end first block); |
| UNSIGNED CHAR GROUP_MODE_0 | (define group mode); |
| UNSIGNED SHORT BEGIN_BLOCK_GROUP_1 | (start second block); |
| UNSIGNED SHORT END_BLOCK_GROUP_1 | (end second block); |
| UNSIGNED CHAR GROUP_MODE_1 | (define group mode); |
| UNSIGNED SHORT BEGIN_BLOCK_GROUP_2 | (start third block); |
| UNSIGNED SHORT END_BLOCK_GROUP_2 | (end third block); |

UNSIGNED CHAR GROUP_MODE_2 (define group mode);

UNSIGNED SHORT BEGIN_BLOCK_GROUP_3 (start fourth block);

UNSIGNED SHORT END_BLOCK_GROUP_3 (end fourth block);

UNSIGNED CHAR GROUP_MODE_3 (define group mode); and

UNSIGNED CHAR LRC (checksum).

The possible replies from the tag 100 are:

UNSIGNED CHAR Command_Accepted = ACK; and

UNSIGNED CHAR Command_Not_Accepted = NAK.

- **Encrypt Random Number**

The encrypt random number command requests the tag 100 to encrypt the POS device random number (CRN) sent to the tag 100 with the tag's main tag key. Preferably, the tag 100 replies by sending the encrypted random number (ECRN) and the tag random number (TRN) generated by the tag 100. Generation of the tag random number (TRN) by the tag 100 is preferably triggered upon receipt of the encrypt random number command from the POS device 200. For maximum security, the tag random number (TRN) is not mathematically related to the POS device 200 random number (CRN). Again, the tag random number (TRN) is used to generate a session key (SK) in the tag 100 and at the host to encrypt and decrypt data.

A typical command protocol sequence for encrypting the POS random number (CRN) follows:

| | |
|---|---|
| UNSIGNED CHAR Encrypt_Random_Number = E4 | (command to encrypt CRN); |
| UNSIGNED CHAR Tag_Random_Number[8] | (generate TRN); and |
| UNSIGNED CHAR LRC | (checksum). |

The possible replies from the tag 100 are:

UNSIGNED CHAR Command_Not_Accepted = NAK; or

UNSIGNED CHAR Command_Accepted = ACK;

| | |
|---|---|
| UNSIGNED CHAR Currrent_Sequence_Number | (increment sequence number); |
| UNSIGNED CHAR Encrypted_Random_Number[8] | (send encrypted CRN [ECRN]); |
| UNSIGNED CHAR Tag_Random_Number[8] | (send TRN); and |
| UNSIGNED CHAR LRC | (checksum). |

- **Secure Write Data**

The secure write data command causes the tag cryptography electronics 104 to store data that is part of this command. The block number in this command is relative to a group and starts from block 0.

The following protocol command sequence securely writes data to a tag 100 according to Figure 9:

| | |
|---|---|
| UNSIGNED CHAR Secure_Write_Data = E5 | (command to write data to tag); |
| UNSIGNED CHAR Sequence_Number | (increment sequence number); |
| UNSIGNED CHAR Group_Number | (select group to be written); |
| UNSIGNED SHORT Block_Number | (select block to be written in group); |
| UNSIGNED CHAR Data[8] | (send 8 bytes of encrypted data); |
| UNSIGNED CHAR Authenticator_Block[8] | (send 8 bytes of encrypted authenticator data); and |
| UNSIGNED CHAR LRC | (checksum). |

The possible replies from the tag 100 are:

UNSIGNED CHAR Command_Accepted = ACK; and

UNSIGNED CHAR Command_Not_Accepted = NAK.

**- Secure Add Data**

The secure add data command causes the tag cryptography electronics 104 to add data that is part of this command to the data that is already stored in the tag cryptography electronics 104. The sequence number sent from the host 300 is one higher than the sequence number stored in the tag cryptography electronics 104. The Add function adds the data in the block to the data in the command and stores the result in the block. The addition is performed in binary coded decimal with carries across bytes. The carry from the most

significant byte (MSB) is dropped. If the addition would result in a carry out of the last byte, then the addition is not performed. Note that the block number in this command is relative to a group and starts from block 0.

A sample protocol command sequence follows:

| | |
|---|---|
| UNSIGNED CHAR Secure_Add_Data = E6 | (send command); |
| UNSIGNED SHORT New_Sequence_number | (send new sequence number); |
| UNSIGNED CHAR Group_Number | (select group); |
| UNSIGNED SHORT Block_Number | (select block); |
| UNSIGNED CHAR Data[8] | (send 8 byte encrypted data); |
| UNSIGNED CHAR Authenticator_Block[8] | (send 8 byte encrypted authenticator data; and |
| UNSIGNED CHAR LRC | (checksum). |

The possible replies from the tag 100 are:

UNSIGNED CHAR Command_Accepted = ACK; and

UNSIGNED CHAR Command_Not_Accepted = NAK.

**- Secure Subtract Data**

The secure subtract data command causes the tag cryptography electronics 104 to perform a subtraction from the secure data storage area in the tag cryptography electronics

104. Preferably, the sequence number is one higher than the sequence number stored in the tag cryptography electronics 104. The subtract function subtracts the number in the command from the number in the block and stores the result in the block. The subtraction is in binary format with borrows across bytes. If the subtraction results in a borrow from the last byte, then the subtraction is performed. Note that the block number in this command is relative to a group and starts from block 0.

A command protocol sequence follows:

| | |
|---|---|
| UNSIGNED CHAR Subtract_Data = E7 | (send command); |
| UNSIGNED SHORT New_Sequence_Number | (send new sequence number); |
| UNSIGNED CHAR Group_Number | (select group); |
| UNSIGNED SHORT Block_Number | (select block); |
| UNSIGNED CHAR Data[8] | (send encrypted data); |
| UNSIGNED CHAR Authenticator_Block[8] | (send encrypted authenticator); |

and

UNSIGNED CHAR LRC.

The possible replies from the tag 100 are:

UNSIGNED CHAR Command_Accepted = ACK; and

UNSIGNED CHAR Command_Not_Accepted = NAK.

- **Secure Read Data**

This commands returns data stored in a secure read memory area of the tag cryptography electronics 104 to the POS device 200. Generally, the data size is assumed to be one block or 8 bytes. Note that the block number in this command is relative to a group and starts from block 0.

A command protocol sequence follows:

UNSIGNED CHAR Read_Data = E8 (send command);

UNSIGNED CHAR Group_Number (select group);

UNSIGNED SHORT Block_Number (select block); and

UNSIGNED CHAR LRC (checksum).

The possible replies from the tag 100 are:

UNSIGNED CHAR Command_Not_Accepted = NAK; or

UNSIGNED CHAR Command_Accepted = ACK;

UNSIGNED CHAR Current_Sequence_Number (send current sequence number);

UNSIGNED CHAR DES_Key_Counter (send DES key counter value);

UNSIGNED CHAR Data[8] /*encrypted*/ (send encrypted data);

UNSIGNED CHAR Authenticator_Block[8] (send encrypted authentic or data);

/*encrypted*/ and

| | |
|---|---|
| UNSIGNED CHAR LRC | (checksum). |

- **Unsecure Write Data**

The unsecure write data command causes the tag cryptography electronics 104 to store data that is part of this command. Note that the block number in this command is relative to a group and starts from block 0.

A command protocol sequence follows;

| | |
|---|---|
| UNSIGNED CHAR Secure_Write_Data = E9 | (send command); |
| UNSIGNED CHAR Sequence_Number | (send sequence number); |
| UNSIGNED CHAR Group_Number | (select group); |
| UNSIGNED SHORT Block_Number | (select block); |
| UNSIGNED CHAR Data[8] | (send data); and |
| UNSIGNED CHAR LRC | (checksum). |

The possible replies from the tag 100 are :

UNSIGNED CHAR Command_Accepted = ACK; and

UNSIGNED CHAR Command_Not_Accepted = NAK.

- **Unsecure Add Data**

The unsecure add data command causes the tag cryptography electronics 104 to add data that is part of this command to the data that is already in storage in the tag cryptography electronics 104. The sequence number is one higher than the sequence number stored in the tag cryptography electronics 104. The Add function adds the data in the block to the data in the command and stores the result in the block. The addition is in binary coded decimal format with carries across bytes. Any carries from the MSBs is lost. If the addition would result in a carry out of the last byte, then the addition will not be performed. Note that the block number in this command is relative to a group and starts from block 0.

The command protocol sequence would be:

| | |
|---|---|
| UNSIGNED CHAR Secure_Add_Data = EA | (send command); |
| UNSIGNED SHORT New_Sequence_number | (send new sequence number); |
| UNSIGNED CHAR Group_Number | (select group); |
| UNSIGNED SHORT Block_Number | (select block); |
| UNSIGNED CHAR Data[8] | (send data); and |
| UNSIGNED CHAR LRC | (checksum). |

The possible replies from the tag 100 are:

UNSIGNED CHAR Command_Accepted = ACK; and

UNSIGNED CHAR Command_Not_Accepted = NAK.

**- Unsecure Subtract Data**

The unsecure subtract data command causes the tag cryptography electronics 104 to perform a subtraction from the data stored in the tag cryptography electronics 104. The sequence number is one higher than the sequence number stored in the tag cryptography electronics 104. The data is not secured, and there are no security checks or encryption on this command. The Subtract function subtracts the number in the command from the number in the block and stores the result in the block. The subtraction is in binary coded decimal format with borrows across bytes. If the subtract would result in a borrow from the last byte, then the subtraction is performed. Note that the block number in this command is relative to a group and starts from block 0.

A command protocol sequence follows:

| | |
|---|---|
| UNSIGNED CHAR Subtract_Data = EB | (send command); |
| UNSIGNED SHORT New_Sequence_Number | (send new sequence number); |
| UNSIGNED CHAR Group_Number | (select group); |
| UNSIGNED SHORT Block_Number | (select block); |
| UNSIGNED CHAR Data[8] | (send data); and |
| UNSIGNED CHAR LRC | (checksum). |

The possible replies from the tag 100 are:

UNSIGNED CHAR Command_Accepted = ACK; or

UNSIGNED CHAR Command_Not_Accepted = NAK.

- **Unsecure Read Data**

The unsecure read data command returns data stored in the tag cryptography electronics 104 to the POS device 200. The data size is assumed to be one block or 8 bytes. Note that the block number in this command is relative to a group and starts from block 0.

A command protocol sequence follows:

| | |
|---|---|
| UNSIGNED CHAR Read_Data = EC | (send command); |
| UNSIGNED SHORT Block_Number | (select block); and |
| UNSIGNED CHAR LRC | (checksum). |

The possible replies from the tag 100 are:

UNSIGNED CHAR Command_Not_Accepted = NAK; or

UNSIGNED CHAR Command_Accepted = ACK;

| | |
|---|---|
| UNSIGNED CHAR Current_Sequence_Number | (send current sequence number); |
| UNSIGNED CHAR DES_Key_Counter | (send DES key counter); |
| UNSIGNED CHAR Data[8] | (send data); and |
| UNSIGNED CHAR LRC | (checksum). |

- **Read DES Module Status**

The read DES module status command reads the tag status registers and also returns the tag cryptography electronics 104 version number.

## CRYPTOGRAPHY MEMORY ORGANIZATION

The tag user memory 120 is preferably configured to include numeric counters or registers. Each counter or register is called a block. Generally, each block is 8 bytes long. There are x blocks of memory in the tag depending on the CPU selected. Data is stored in the blocks in Big Endian mode, that is, the lowest address is the most significant byte of data. As noted, the tag user memory 120 is divided into four different areas called groups. The commands available for each group depend on the content of the Group Mode register for that group. It is permissible for the group begin block and end block numbers to overlap. In these instances, groups can share blocks and the data stored therein.

Add and subtract operations are performed in binary coded decimal (BCD). Typically, each 8 byte block is capable of storing a maximum of 16 decimal numbers. If there are x number of bytes in the user memory 120, then there are x/8 blocks of data. Blocks are numbered from 0 to (x/8) - 1. A detailed example is given below on block divisions and possible uses of the groups.

In the following example, there is 1024 bytes of user memory in the cryptography electronics 104 assuming there are 128 memory blocks. The group registers would be setup as follows:

```
UNSIGNED SHORT BEGIN_BLOCK_GROUP_0     = 0x0000;
UNSIGNED SHORT END_BLOCK_GROUP_0       = 0x001f;
UNSIGNED CHAR GROUP_MODE_0             = 0x0f
UNSIGNED SHORT BEGIN_BLOCK_GROUP_1     = 0x0020;
UNSIGNED SHORT END_BLOCK_GROUP_1       = 0x003f;
UNSIGNED CHAR GROUP_MODE_1             = 0x87
UNSIGNED SHORT BEGIN_BLOCK_GROUP_2     = 0x0040;
UNSIGNED SHORT END_BLOCK_GROUP_2       = 0x005f
UNSIGNED CHAR GROUP_MODE_2             = 0xc3
UNSIGNED SHORT BEGIN_BLOCK_GROUP_3     = 0x0060;
UNSIGNED SHORT END_BLOCK_GROUP_3       = 0x007f;
UNSIGNED CHAR GROUP_MODE_3             = 0x81
```

With reference to Figures 10 and 11 and the example setup, the first group of blocks is composed of the first 32 blocks in the user memory, numbered from 0 to 31. An attempt to perform an unsecure read or subtract function to these blocks will not be accepted. The data is preferably stored as clear text data. A different session key (SK) is used to manipulate the data at each session rather than expose the same key to constant scrutiny.

The second group of blocks is composed of 32 blocks numbered from 32 to 63. This group of blocks accepts an unsecure read function. An attempt to perform an unsecure add or subtract operation to one of these blocks will not be accepted. This group of blocks is preferably used as a cash storage device and other applications analogous to prepaid cards or smartcards.

The third group of blocks is composed of 32 blocks numbered from 64 to 95. This group of blocks may accept both an unsecure read function and an unsecure subtract function. Note that secure write and add functions are required. This group of blocks is preferably a data storage location for loyalty points. A local source 32 (or even the POS device 200 or local station) providing benefits in exchange for the loyalty points can read and subtract from data stored in this group, but is not able to write or add to the data stored in this group.

The last group of blocks is composed of 32 blocks numbered from 96 to 127 and will accept an unsecure read function. An attempt to perform an add or subtract function, either secure or unsecure, will not be accepted. The write operation must be secure. A block configured in this manner is useful for storing customer information on the tag 100 which is readily accessible by sources other than the host 300. The write operation is secure to prevent unauthorized sources, typically those other than the host 300, from altering customer

information. Figures 10 and 11 include preferable commands and the recommended usage for each group in this example.

## USER MEMORY ORGANIZATION

In an active tag 100, that is one having an internal source of power, user memory 120 is preferably 256 bytes long. The memory 120 is preferably divided into three different partitions as allowed by the three different sets of partition descriptor registers. Each partition requires a different password for various types of access by various sources. Figure 12 illustrates possible partitions for the tag 100 and outlines the recommended requirements for the respective passwords for the different partitions. Similar setup is available for passive tag and is limited only to the communication technology's ability to operate the tags control and memory structure.

The various partitions and the associated passwords provide varying types of access for different sources adapted to read, write or modify data in the tag 100. Preferably, there are three passwords, SuperUser, Administrator and User. The SuperUser password is known only by the host 300. Each tag 100 will have a different SuperUser password, which is calculated from the tag ID. The first partition, partition 0, associated with the SuperUser password may include customer definition data fields as shown in Figures 13A and 13B. Partition 0 can be read by local sources without the SuperUser password. However, only the

host 300, using the SuperUser password, may write data to this partition or otherwise modify data therein.

The host network 300 may also generate a host authentication code from the tag ID. The SuperUser password and host authentication code will be different for each tag 100 and may be used during authorization or secure data transfer between the tag 100 and the host 300 when DES encryption or the primary cryptography technique is unavailable or unnecessary. The authentication code simply provides another security feature for data transfer between the tag 100 and host 300.

The second partition, partition number 1, allows the host 300, POS device 200 or other local sources to access and modify data in the tag 100. The second partition preferably includes local and host authentication codes. The POS device 200 can calculate the administrator password from the tag ID. The local authentication code is used for authorization for the various local sources, including the POS device 200, whereas the host authentication code is sent to the host for authentication. For example, the POS device 200 can calculate the administrator password for partition number 1 from the tag ID and then read the local and host authentication code. If local transactions are desired, the POS device 200 checks the local authentication code and authorizes a local transaction accordingly. If a host related transaction is necessary, the host authentication code is passed on to the host for authentication.

Preferably, an administrative DES key is used to generate the administrator password for the second partition (partition number 1). Yet another DES key is used to generate the local authentication code. Thus, the administrator password and the local authentication code is generated with additional DES keys and the tag ID. The administrator Partition

password and the local authentication code should be unique for each tag. Figure 14 provides a typical layout for the second partition, (partition number 1).

The third partition, partition number 2, provides a scratch pad for any of the various sources communicating with the tag 100. The scratch pad provides complete read and write access for any of the local sources. The scratch pad can be used for virtually any type of application where security is not of utmost importance. Notably, this portion of memory allows virtually any source to communicate with the tag, especially those without cryptography capabilities. The scratch pad feature expands tag compatibility with virtually any type of source having remote intelligent communication capability. Preferably, the scratch pad area allows a local system to save information to be read a short time later at the same location or station. Typically, the scratch pad is 32 bytes long, and its organization of data fields, if any, is determined by the local system. A User password may be used to limit access to those having a password and provide modest security. The User password for the user partition is preferably the same in all tags. Typically, the POS device operator or tag issuer will supply the password. Figure 15 shows a typical layout of the third partition, (partition number 2).

## PREFERRED PASSWORD AND AUTHENTICATION CODE GENERATION

The SuperUser password for the SuperUser partition number 0 is generated by the host 300 and is preferably unknown to the local sources, including the POS device 200. The SuperUser password is different for each tag and is based on the tag ID. A preferred method of generating this password is shown in Figure 16.

In order to generate the SuperUser password the least significant 8 bytes of the tag ID are XORed with an 8 byte constant chosen by the host 300. The result is then XORed with a unique host password generation key, DES encrypted with the host password generation key and XORed again with the host password generation key. The result of this is an 8 byte value. This 8 byte value is then split into two parts consisting of the least significant 4 bytes (lower half) and the most significant 4 bytes (upper half). These two halves are XORed together and then ANDed with the 4 byte hexadecimal constant 7F FF FF FF. This has the effect of making the most significant bit of the 32 bit value equal to a logical zero. The result of this operation is the tag's SuperUser Password that is injected and stored in to tag 100 during initialization.

The administrator password is generated by the host 300 at the time the tag is initialized. Each tag gets a different administrator password, which is calculated from the tag ID. The generation process uses a unique administrator DES key, preferably supplied by the POS device or dispenser provider. The administrator password generation process is shown in Figure 17. The administrator password is preferably generated by the POS device 200 to allow read access to the local and host authentication codes stored in partition number 1.

In order to generate the administrator password the least significant 8 bytes of the tag ID are first XORed with the POS provider password generation key, then DES encrypted with the POS provider password generation key, and then XORed again with the POS provider password generation key. The result of this is an 8 byte value. The 8 byte value is then split into two parts consisting of the least significant 4 bytes (lower half) and the most significant 4 bytes (upper half). These two halves are XORed together and then ORed with

the 4 byte hexadecimal constant 80 00 00 00. This has the effect of making the most significant bit of the 32 bit value equal to a logical one. The result of this operation is the tag administrator password that is injected and stored in the tag.

The host authentication code is read by the POS device 200 from partition 1, the administrator partition. The code is transmitted to the host 300 to authenticate the tag. Each tag has a different authentication code, which is generated from the tag ID. A preferred method of generating this code is shown in Figure 8.

In order to generate the host authentication code the least significant 8 bytes of the tag ID are XORed with an 8 byte constant chosen by the host. The result is then XORed with a unique host authentication code generation key, DES encrypted with the host authentication code generation key, and XORed again with the host authentication code generation key. The result of this is an 8 byte value. This 8 byte value is stored in the administrator partition of the tag 100 as the host authentication code.

The local authentication code is also generated by the host 300 at the time the tag is initialized. Each tag 100 gets a different local authentication code that is calculated from the tag ID. Preferably, the generation process uses a DES key supplied by POS device provider for this purpose. The local authentication code Generation Process is shown in Figure 19. The local authentication code can be generated by the POS device to allow the local system to authenticate the tag.

In order to generate the local authentication code the least significant 8 bytes of the tag ID are first XORed with the local authentication code generator DES key, then DES encrypted with the local authentication code generator DES key, and then XORed again with

the local authentication code generator key. The result is an 8 byte value that is the local authentication code. The local authentication code is stored in the tag's administrator partition.

Notably, the convoluted processes disclosed herein to determine and calculate the various encryption keys, codes, and numbers, such as the ID number, are for example only. The combination of cryptography and logical operations may be reduced, amplified or modified to provide the desired results.

Certain modifications and improvements will occur to those skilled in the art upon a reading of the foregoing description. It should be understood that all such modifications and improvements have been deleted herein for the sake of conciseness and readability but are properly within the scope of the following claims.

CLAIMS:

1. A system for providing secure transactions between a host, a point-of-sale device and a remote transponder, wherein the transponder and host are arranged to communicate information and/or instructions between each other via the point-of-sale device, characterised in that at least one of the host or transducer authenticates that the information and/or instruction received originated from the other, the authentication being based on data available to the host and transponder which data is not available to the point-of-sale device.

2. A system as claimed in claim 1 providing remote transactions with a plurality of sources comprising:

communication electronics for providing wireless, bi-directional remote communications with a point-of-sale device and a host through the point-of-sale device;

controller electronics associated with said communication electronics;

memory associated with said electronics and divided into at least first and second partitions, and

said controller requiring receipt of a first password from the host, through the point-of-sale device, to alter content of said first partition and receipt of a second password from the point-of-sale device to alter content of said second partition.

3. A system as claimed in claim 2 wherein said communication electronics is further adapted to remotely communicate with local sources and said memory includes a third partition, said controller providing access to said third partition by the local sources to read, write or otherwise alter content in said third partition.

4. A system as claimed in claims 2 or 3 wherein said first and/or second password is unique to said remote communication unit.

5. A system as claimed in any one of claims 2 to 4 wherein said first partition holds a host authentication code used by the host to authenticate said remote communications unit for transactions with the host and a local authentication code used by the point-of-sale device to authenticate said remote communications unit for transactions with said point-of-sale device.

6. A system as claimed in any one of claims 2 to 5 wherein said controller electronics is adapted to encrypt and decrypt data transmitted from and received by said communication electronics.

7. A system as claimed in claim 1 wherein:

(a) the transponder has security electronics providing high-level cryptography and lower-level security, said security electronics operatively associated with communication electronics, which provides bi-directional remote communications;

(b) the transponder is adapted to transmit data encrypted with said high-level cryptography to said point-of-sale device;

(c) said point-of-sale device is adapted to transmit said data encrypted with said high level cryptography to the host and receive first data from the host and transmit the first data to said transponder; and

(d) said transponder is further adapted to transmit second data to said point-of-sale device according to said lower-level security for use at said point-of-sale device.

8. A system as claimed in claim 7 wherein said lower level security includes a password stored in memory and is adapted to transmit data from said transponder upon receipt of a transmission including the password.

9. A system as claimed in claim 7 or 8 wherein said lower-level security is adapted to facilitate transmitting and receiving data to and from remote sources other than said point-of-sale device.

10. A system as claimed claims 7, 8 or 9 wherein said high-level cryptography uses a main cryptography key stored in said transponder and the host, said cryptography key not being stored in said point-of-sale device and used to provide secure transactions between said transponder and the host through said point-of-sale device.

11. A system as claimed in any one of claims 7 to 10 further comprising the features of any one of claims 2 to 6.

12. A system as claimed in claim 1 wherein the transponder and host communicate with each other through the point-of-sale using a protocol, said protocol from the host to transponder through the point-of-sale device comprising:

a communication block including:

a a command to alter data in a memory of the transponder, the memory divided into groups and the groups divided into blocks;

b a host sequence number corresponding to a transponder sequence number stored in the transponder, said sequence number indicative of the number of operations the host has performed with the transponder;

c a memory group identifier for selecting the memory group to be operated on;

d a memory block identifier for selecting the memory block to be operated on within the memory group;

e host data to be used to alter the data in the selected memory block according to the command; and

f host authentication data, said host authentication data formed by a convoluted operation on the host data and used to authenticate the host data at the transponder; and

a communication check based on at least a portion of the communication block.

13. A system as claimed in claim 12 wherein the command of the communication block is add, subtract or store, and said command is interpreted by the transponder to add to, or

subtract from the data in the selected memory block the host data, or to store the host data in the memory block.

14. A system as claimed in claims 12 or 13 wherein the transponder is arranged to ignore said command of the command block if the host sequence number does not correspond to an expected sequence number based on the transponder sequence number.

15. A system as claimed in claims 12, 13 or 14 wherein the host data and host authentication data are encrypted.

16. A system as claimed in claim 12 wherein the transponder and host network communicate with each other through the point-of-sale device using a protocol, said protocol from the transponder to the host through the point-of-sale device comprising:

a communication block including:

(a) a read command acknowledgment indicating a command from the host to read data in a selected memory block of the transponder has been accepted;

(b) a transponder sequence number stored in the transponder identical to a host sequence number, said sequence number indicative of the number of operations the host has performed with the transponder;

(c) an encryption key counter value storing a value corresponding to how many times a encryption key has been changed in the transponder;

(d) transponder data from the selected memory block in the memory; and

(e) transponder authentication data, said transponder authentication data formed by a convoluted operation on the transponder data and used to authenticate the transponder data at the host; and

a communication check based on at least a portion of the communication block.

17. A system as claimed in claim 16 wherein the transponder data and transponder authentication data are encrypted.

18. A system as claimed in claim 17 wherein the point-of-sale device is unable to decrypt the encrypted transponder data and encrypted transponder authentication data.

19. A system as claimed in claim 1 wherein the transponder remotely communicates with the host via a protocol comprising:

(a) a first transponder to point-of-sale device communication block including a transponder identifier;

(b) a first point-of-sale device to transponder communication block including a random number generated at the point-of-sale device;

(c) a second transponder to point-of-sale device communication block including the random number encrypted;

(d) a first point-of-sale device to host communication block including the transponder identifier, the random number and the encrypted random number; and

(e) a first host to point-of-sale device communication block including a signal authorizing the transponder if the host system authorizes the transponder.

20. A system as claimed in claim 19 further including:

(a) a third transponder to point-of-sale device communication block including a transponder generated random number; and

(b) a second point-of-sale device to host communication block including the transponder generated random number.

21. A system as claimed in claim 20 further including:

(a) a second host to point-of-sale device communication block including a command to write or modify data to the transponder; and

(b) a third point-of-sale device to host communication block including the command to write or modify data to the transponder.

22. A system as claimed in claim 20 further including:

(a) a second host to point-of-sale device communication block including a command to read data from the transponder;

(b) a third point-of-sale device to host communication block including the command to read data from the transponder;

(c) a fourth transponder to point-of -sale device communication block including data from the transponder according to the read command; and

(d) a third point-of-sale device to host communication block including the data read from the transponder.

23. A system as claimed in claim 1 said transponder comprising:

communication electronics for providing wireless, bi-directional, secure, communications with a point-of-sale device that can communicate with a host; and cryptography electronics including a key storage for encrypting with a cryptography key stored in the key storage the remote communications to the host via the point-of-sale device and decrypting the remote communications from the host via the point-of-sale device, said cryptography key being unknown to said point-of-sale device.

24. A system as claimed in claim 23 wherein said cryptography and communication electronics are adapted to encrypt data and transmit the encrypted data to the host via the point-of-sale device.

25. A system as claimed in claims 23 or 24 wherein said cryptography electronics generates a session key different from the cryptography key for encrypting certain of the data transmitted to the point-of-sale device and on to the host.

26. A system as claimed in claim 25 wherein said session key is a function of the cryptography key.

27. A system as claimed in claims 25 or 26 wherein said cryptography and communication electronics operate to transmit a session key generation signal to the host through the point-of-sale device to enable the host to generate the session key.

28. A system as claimed in claim 25 wherein said electronics include a random number generator to generate a random number, said session key being said random number generation signal.

29. A system as claimed in any one of claims 25 to 28 wherein said cryptography electronics uses said session key to decrypt data transmitted to said transponder from the host through the point-of-sale device.

30. A system as claimed in any one of claims 23 to 29 wherein said memory includes a unique transponder identifier and said communication electronics is adapted to transmit said identifier.

31. A system as claimed in any one of claims 25 to 30 wherein said communication electronics are adapted to provide bi-directional communications with local sources in addition to the point-of-sale device.

32. A system as claimed in claim 31 wherein said encryption electronics encrypts the data transmitted to and decrypts the data received from the local sources with a key different from said cryptography key.

33. A system as claimed in claims 31 or 32 wherein said electronics are further adapted to receive a password from the at least one of the local sources to allow access to said memory, said memory providing a scratchpad for the local source to use as desired when communicating with said transponder.

34. A system as claimed in any one of claims 23 to 33 wherein said encryption electronics encrypts data destined for, and decrypts the data originated from, the point-of-sale device with a different crytography key to that used for encrypting data transmitted to the host through the point-of-sale device.

35. A system as claimed in any one of claims 23 to 34 wherein a memory contains transponder identification data, said communication electronics are adapted to receive data from the point-of-sale device, said cryptography electronics are adapted to encrypt the data to form encrypted data, and said communication electronics are adapted to transmit said transponder identification data and said encrypted data wherein said encrypted data and said identification data are ultimately transmitted to the host through the point-of-sale device for authorization.

36. A system as claimed in claim 35 wherein said transponder transmits said encrypted data and said identification data to the point-to-sale device, said data is transmitted to said transponder and the host by the point-of-sale device and the host, and authorization by the host includes encrypting said data to obtain host encrypted data and comparing said encrypted data with the host encrypted data.

37. A system as claimed in any one of claims 23 to 36 wherein said communications electronics are adapted to receive authorization data from the point-of-sale device, said cryptography electronics are adapted to encrypt the received authorization data and said communication electronics are further adapted to transmit said encrypted received authorization data for host authorization.

38. A system as claimed in claim 37 wherein said authorisation data is a random number generated and transmitted to said transponder by the point of sale device.

39. A system as claimed in any one of claims 23 to 36 wherein said transponder further includes a transponder random number generator operatively associated with said electronics and adapted to generate transponder random numbers upon receipt of a signal generated by the point-of-sale device, said cryptography electronics being adapted to encrypt said transponder random number to generate a session key for decrypting data transmitted to said transponder and written to said memory.

40. A system as claimed in claim 39 wherein said communication electronics are adapted transmit said transponder random number generated by said random number generator to said point-of-sale device, which relays said transponder generated random number to the host wherein said host generates a key identical to said session key.

41. A system as claimed in claim 23 comprising a cryptography key generator for generating keys used in remote communication devices comprising:

a remote communication device identification number generator adapted to generate unique identification numbers for a plurality of remote communication devices,

a memory having at least one master encryption key stored therein;

cryptography electronics associated with said host processing system adapted to encrypt a function of said identification number using said at least one master encryption key to provide a main remote communication device key; and

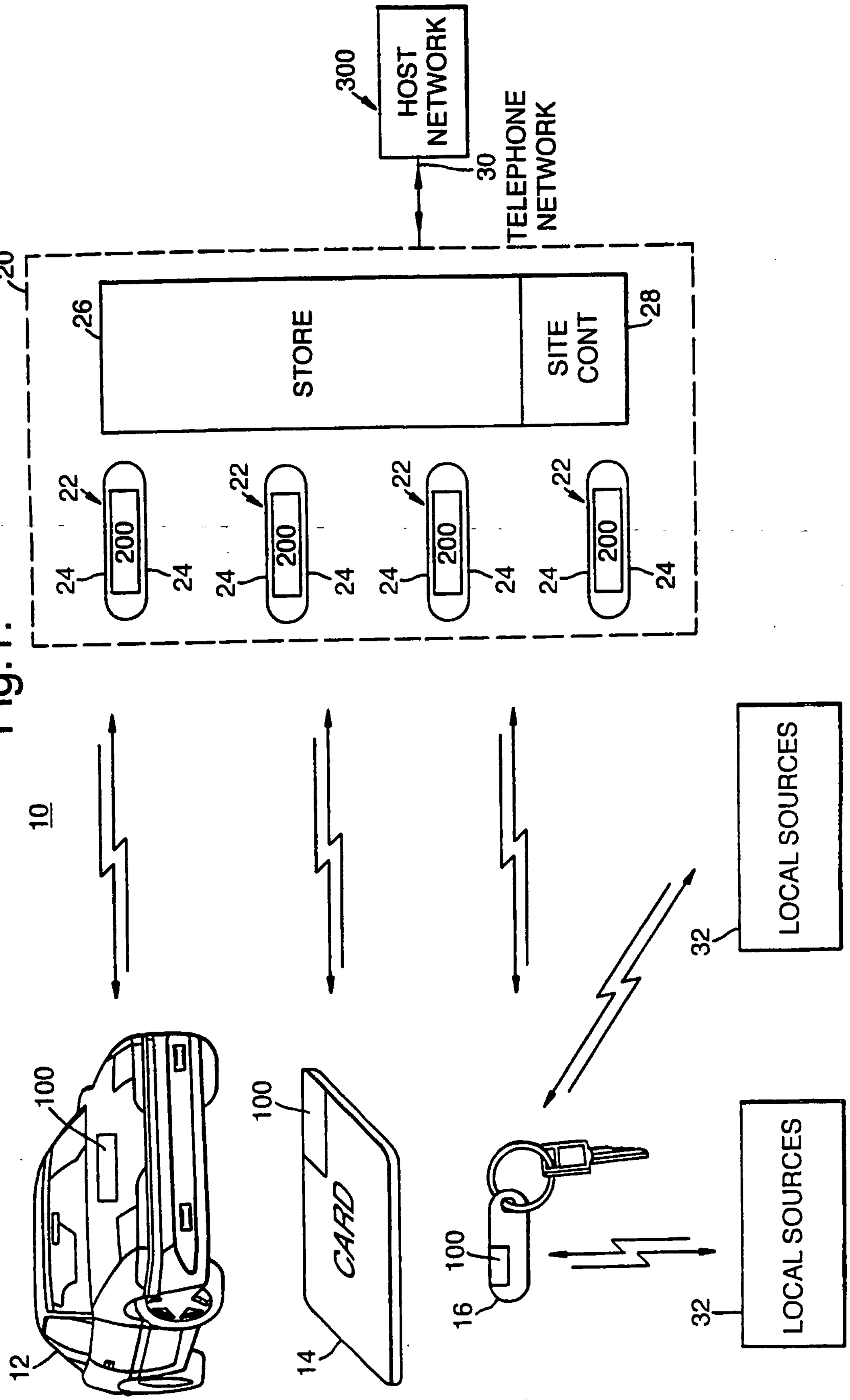
communication electronics associated with said cryptography electronics adapted to transmit said remote communication device identification number and said remote communication device key to a corresponding remote communication device.

42. A system as claimed in any one of claims 1 to 41 wherein said point of sale device is a fuel dispenser.

43. A system as claimed in any one of claims 1 to 42 wherein said transponder is a payment card, tag or token associated with a customer or vehicle.

44. A system as claimed in any one of claims 1 to 43 wherein the host is a network through which a transaction associated with the transponder can be accredited to a particular account.

Fig.1.


10
HOST NETWORK
300
30
TELEPHONE NETWORK
20
STORE
26
SITE CONT
28
22
200
24
100
12
CARD
14
16
32
LOCAL SOURCES

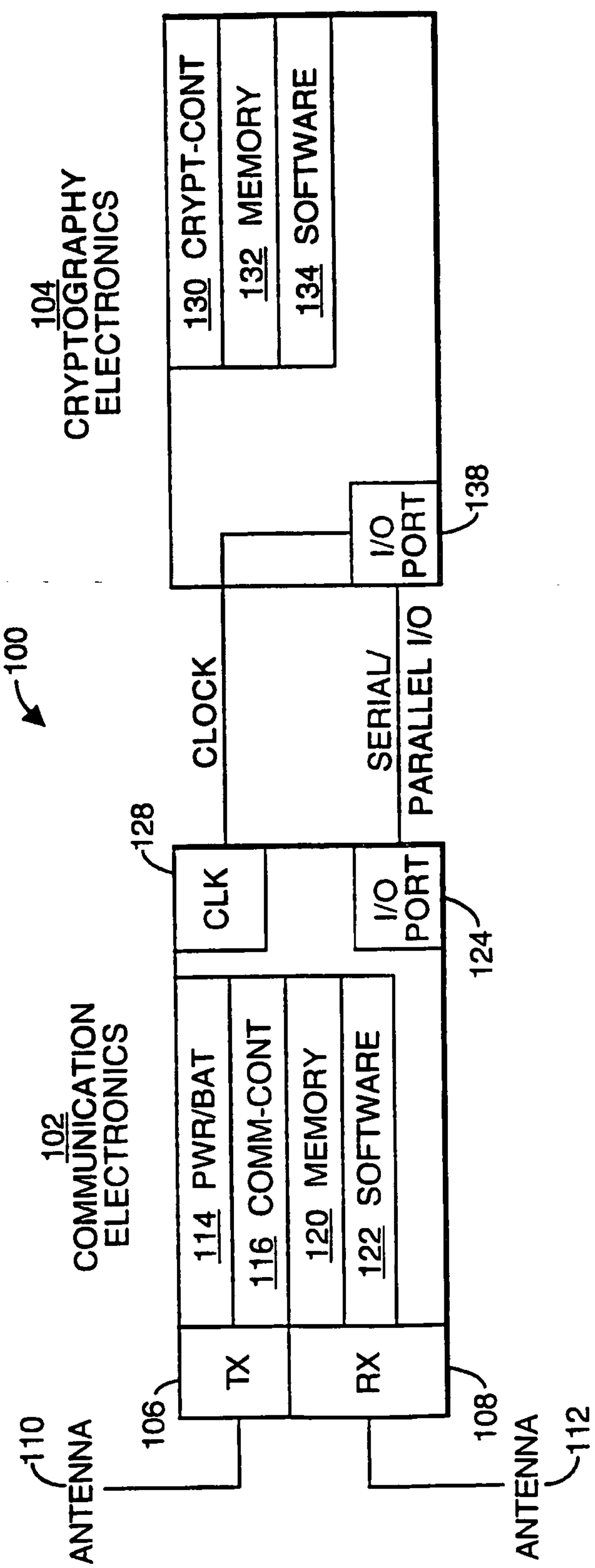

100
104
CRYPTOGRAPHY
ELECTRONICS
130 CRYPT-CONT
132 MEMORY
134 SOFTWARE
I/O
PORT
138
CLOCK
SERIAL/
PARALLEL I/O
128
CLK
I/O
PORT
124
102
COMMUNICATION
ELECTRONICS
114 PWR/BAT
116 COMM-CONT
120 MEMORY
122 SOFTWARE
TX
RX
106
108
ANTENNA
110
ANTENNA
112

Fig.2A.

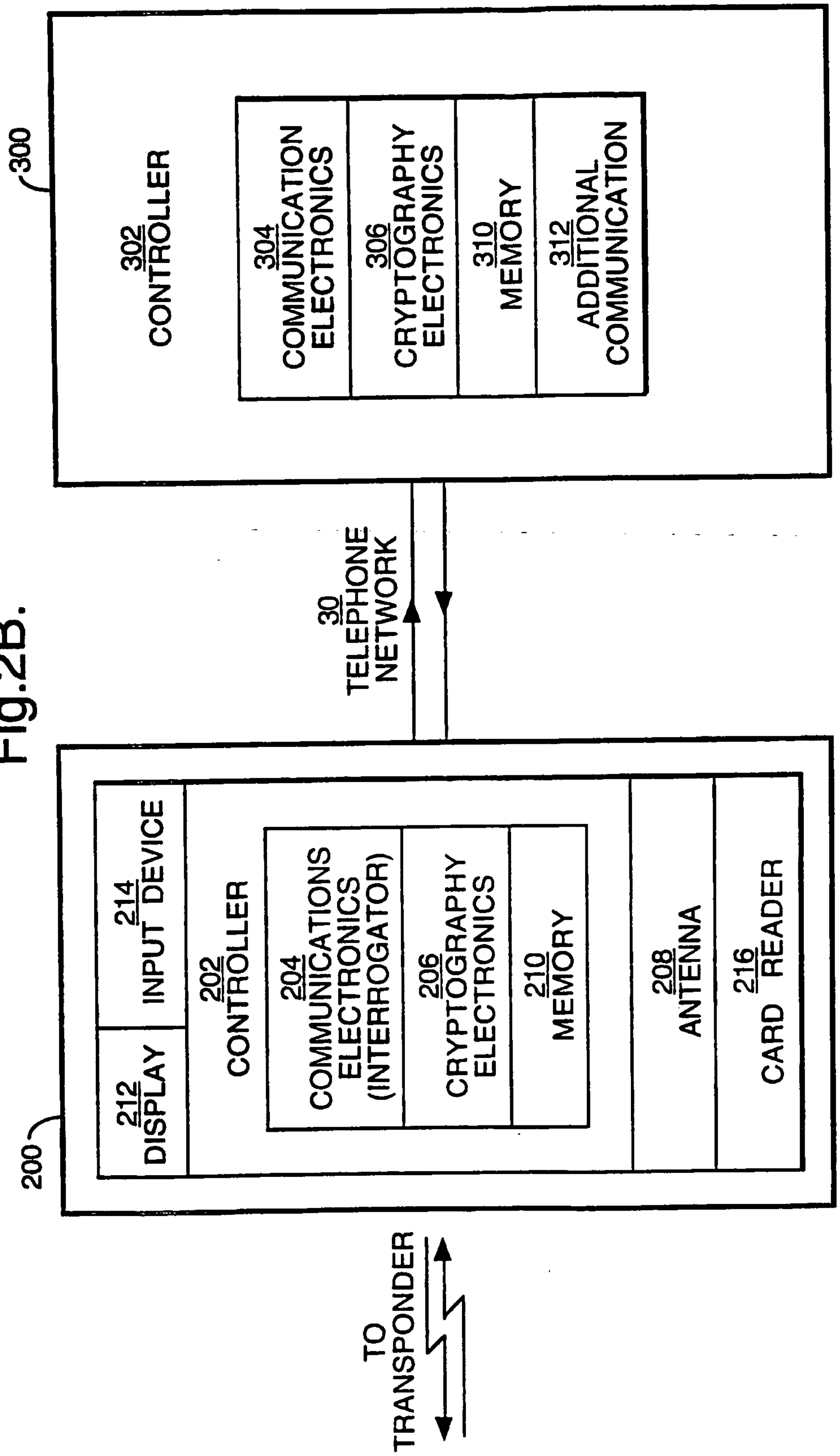

300
302
CONTROLLER
304
COMMUNICATION
ELECTRONICS
306
CRYPTOGRAPHY
ELECTRONICS
310
MEMORY
312
ADDITIONAL
COMMUNICATION
30
TELEPHONE
NETWORK
200
212
DISPLAY
214
INPUT DEVICE
202
CONTROLLER
204
COMMUNICATIONS
ELECTRONICS
(INTERROGATOR)
206
CRYPTOGRAPHY
ELECTRONICS
210
MEMORY
208
ANTENNA
216
CARD READER
TO
TRANSPONDER

Fig.2B.

Fig.2C.

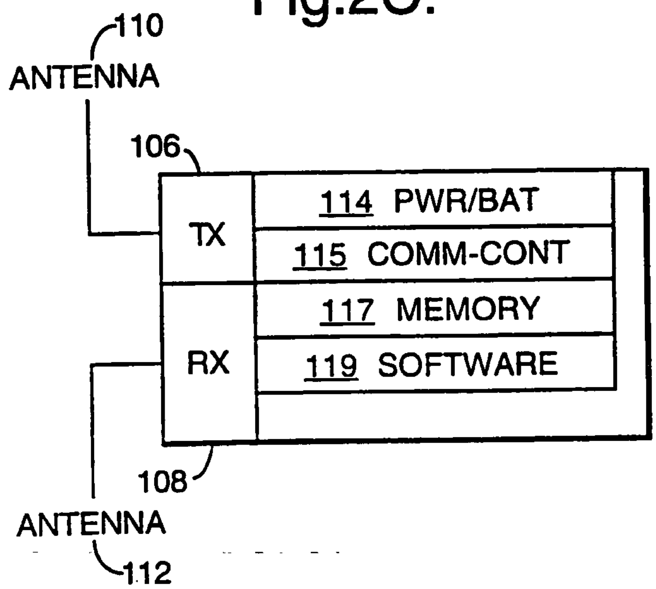



Fig.3A.

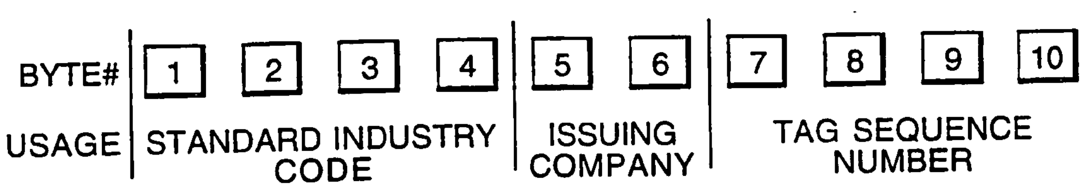
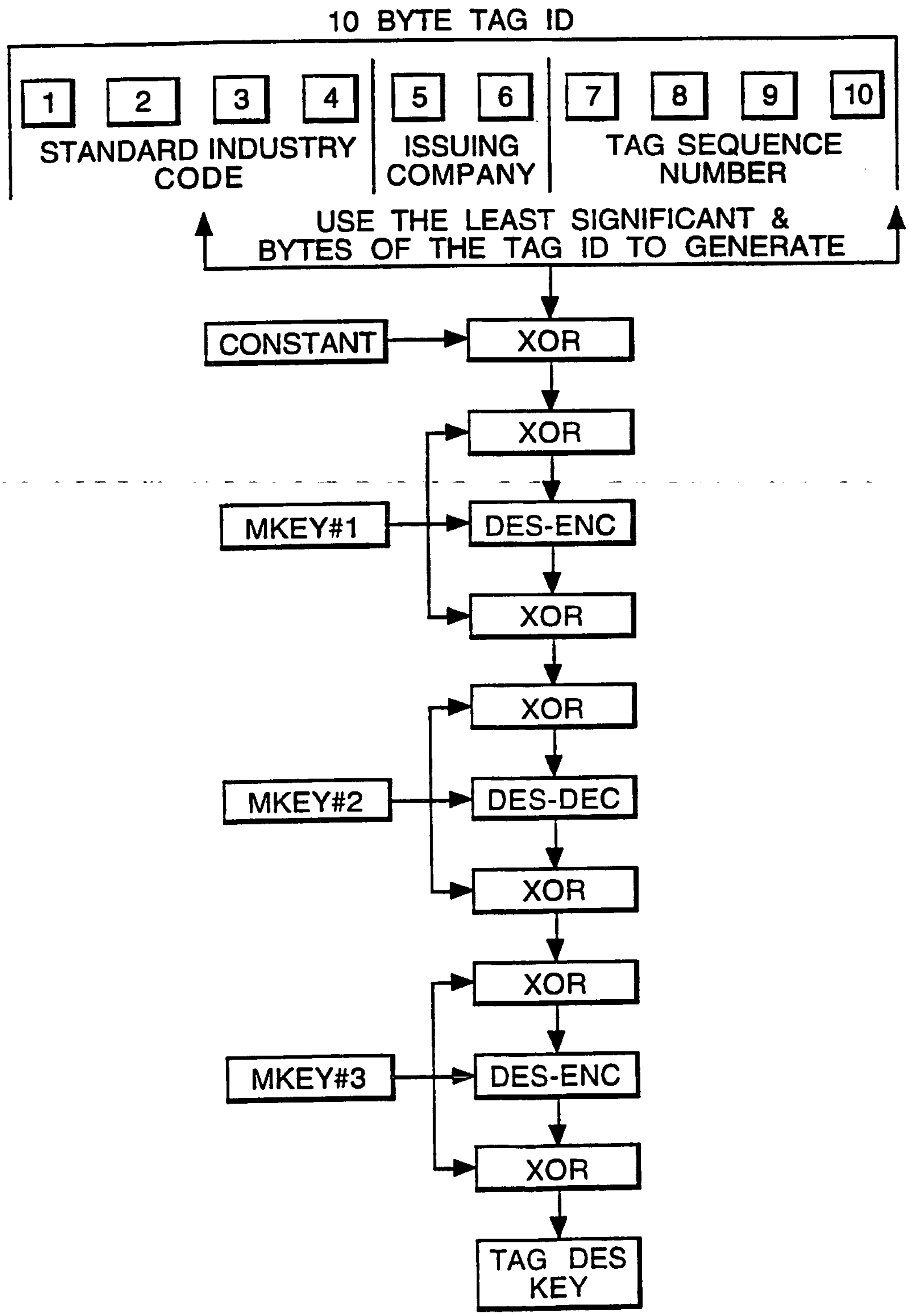



Fig.3B.

| | TAG ID VALUE-BYTES 5 AND 6 | |
|---|---|---|
| **COMPANY NAME** | **ASCII** | **HEX** |
| SHELL | SH | 53,48 |
| LOCAL TEST | 00 | 30,30 |
| NETWORK TEST | 01 | 30,31 |

Fig.4A.


10 BYTE TAG ID
1
2
3
4
5
6
7
8
9
10
STANDARD INDUSTRY CODE
ISSUING COMPANY
TAG SEQUENCE NUMBER
USE THE LEAST SIGNIFICANT & BYTES OF THE TAG ID TO GENERATE
CONSTANT
XOR
XOR
MKEY#1
DES-ENC
XOR
XOR
MKEY#2
DES-DEC
XOR
XOR
MKEY#3
DES-ENC
XOR
TAG DES KEY

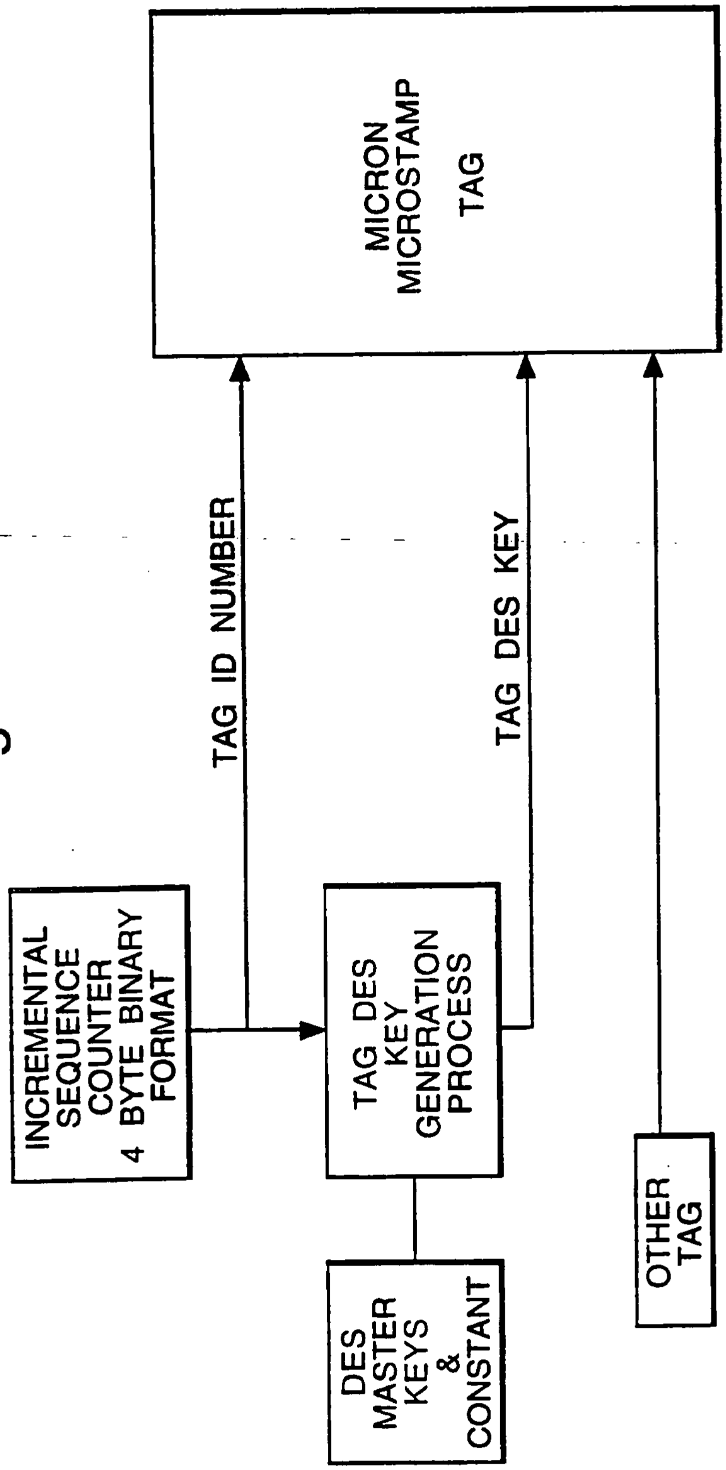

MICRON
MICROSTAMP
TAG
TAG ID NUMBER
TAG DES KEY
INCREMENTAL
SEQUENCE
COUNTER
4 BYTE BINARY
FORMAT
TAG DES
KEY
GENERATION
PROCESS
DES
MASTER
KEYS
&
CONSTANT
OTHER
TAG

Fig.4B.

Fig.5.

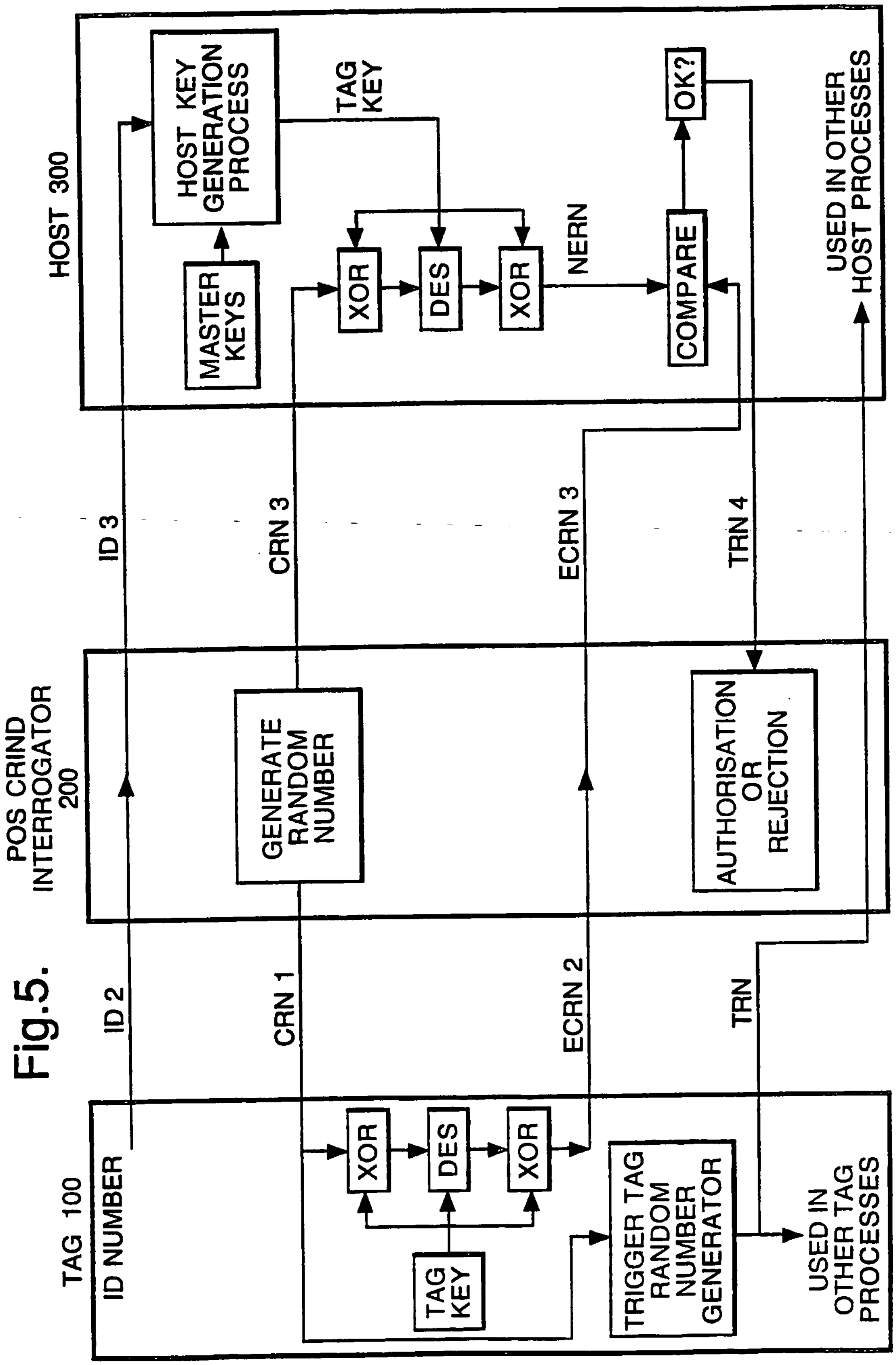

TAG 100
ID NUMBER
XOR
DES
XOR
TAG KEY
TRIGGER TAG RANDOM NUMBER GENERATOR
USED IN OTHER TAG PROCESSES
ID 2
CRN 1
ECRN 2
TRN
POS CRIND INTERROGATOR 200
GENERATE RANDOM NUMBER
AUTHORISATION OR REJECTION
ID 3
CRN 3
ECRN 3
TRN 4
HOST 300
MASTER KEYS
HOST KEY GENERATION PROCESS
TAG KEY
XOR
DES
XOR
NERN
COMPARE
OK?
USED IN OTHER HOST PROCESSES

Fig.6.

| GROUP MODE REGISTER PROGRAMMING | |
|---|---|
| COMMAND NAME | BIT POSITION |
| UNSECURE READ DATA | 7 (MSB) |
| UNSECURE SUBTRACT DATA | 6 |
| UNSECURE ADD DATA | 5 |
| UNSECURE WRITE DATA | 4 |
| SECURE READ DATA | 3 |
| SECURE SUBTRACT DATA | 2 |
| SECURE ADD DATA | 1 |
| SECURE WRITE DATA | 0 (LSB) |

Fig.7.

| GROUP MODE REGISTER | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

Fig.8.

| COMMAND NAME | COMMAND CODE (IN HEX) |
|---|---|
| SET DES PASSWORD | E1 |
| SET DES KEY | E2 |
| SET GROUP REGISTERS | E3 |
| ENCRYPT RANDOM NUMBER | E4 |
| SECURE WRITE DATA | E5 |
| SECURE ADD DATA | E6 |
| SECURE SUBTRACT DATA | E7 |
| UNSECURE READ DATA | E8 |
| UNSECURE WRITE DATA | E9 |
| UNSECURE ADD DATA | EA |
| UNSECURE SUBTRACT DATA | EB |
| UNSECURE READ DATA | EC |
| READ DES MODULE STATUS | ED |

Fig.10.

| | AVAILABLE FUNCTION | | | | |
|---|---|---|---|---|---|
| GROUP# | WRITE | READ | ADD | SUBTRACT | GROUP FUNCTION |
| GROUP#0 | SECURE | SECURE | SECURE | SECURE | TBD |
| GROUP#1 | SECURE | UNSECURE | SECURE | SECURE | CASH OR PREPAID TAG |
| GROUP#2 | SECURE | UNSECURE | SECURE | UNSECURE | LOYALTY TAG |
| GROUP#3 | SECURE | UNSECURE | N/A | N/A | CUSTOMER INFO |

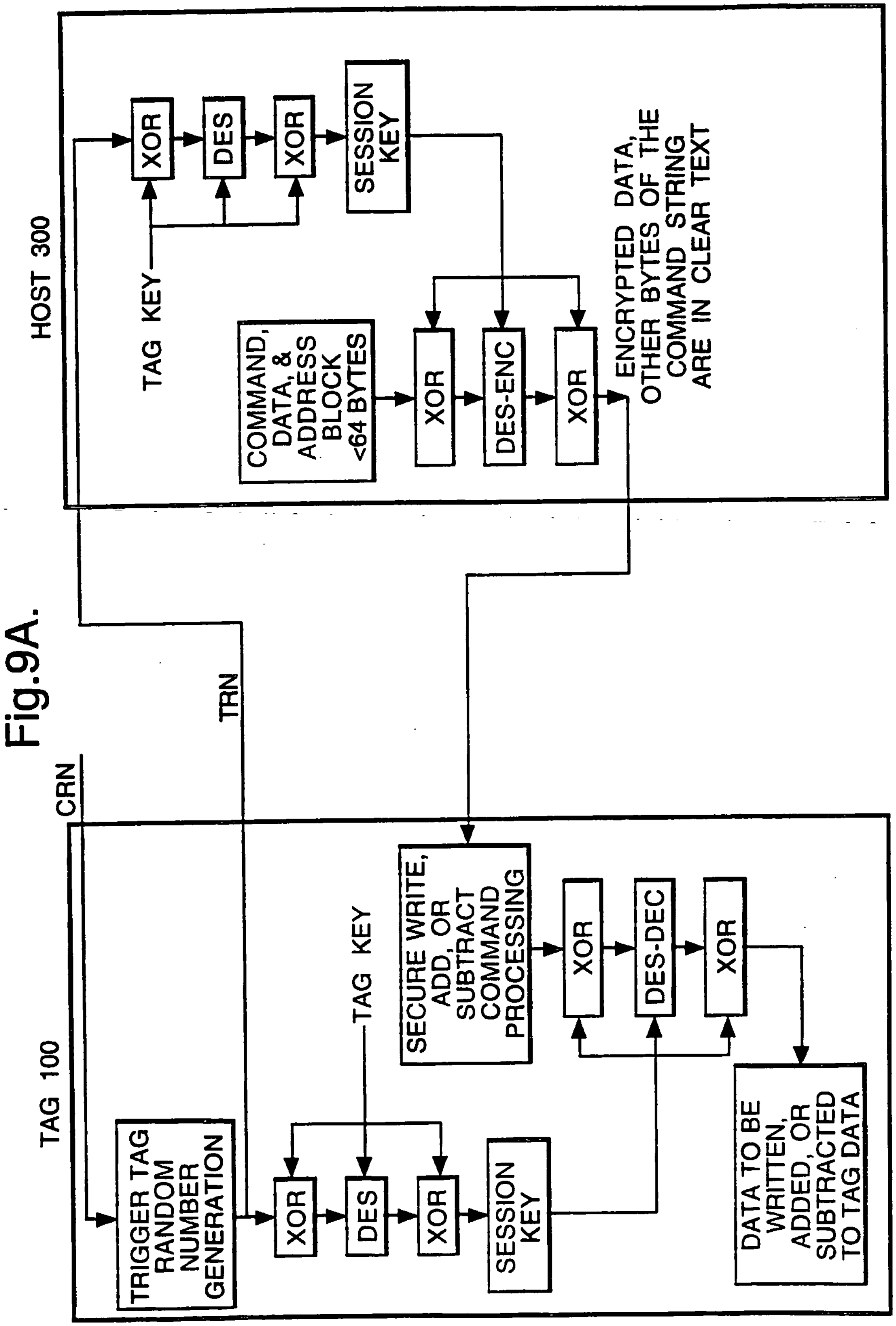

HOST 300
TAG KEY
XOR
DES
XOR
SESSION KEY
COMMAND, DATA, & ADDRESS BLOCK <64 BYTES
XOR
DES-ENC
XOR
ENCRYPTED DATA, OTHER BYTES OF THE COMMAND STRING ARE IN CLEAR TEXT
TRN
CRN
TAG 100
TRIGGER TAG RANDOM NUMBER GENERATION
TAG KEY
XOR
DES
XOR
SESSION KEY
SECURE WRITE, ADD, OR SUBTRACT COMMAND PROCESSING
XOR
DES-DEC
XOR
DATA TO BE WRITTEN, ADDED, OR SUBTRACTED TO TAG DATA

Fig.9A.

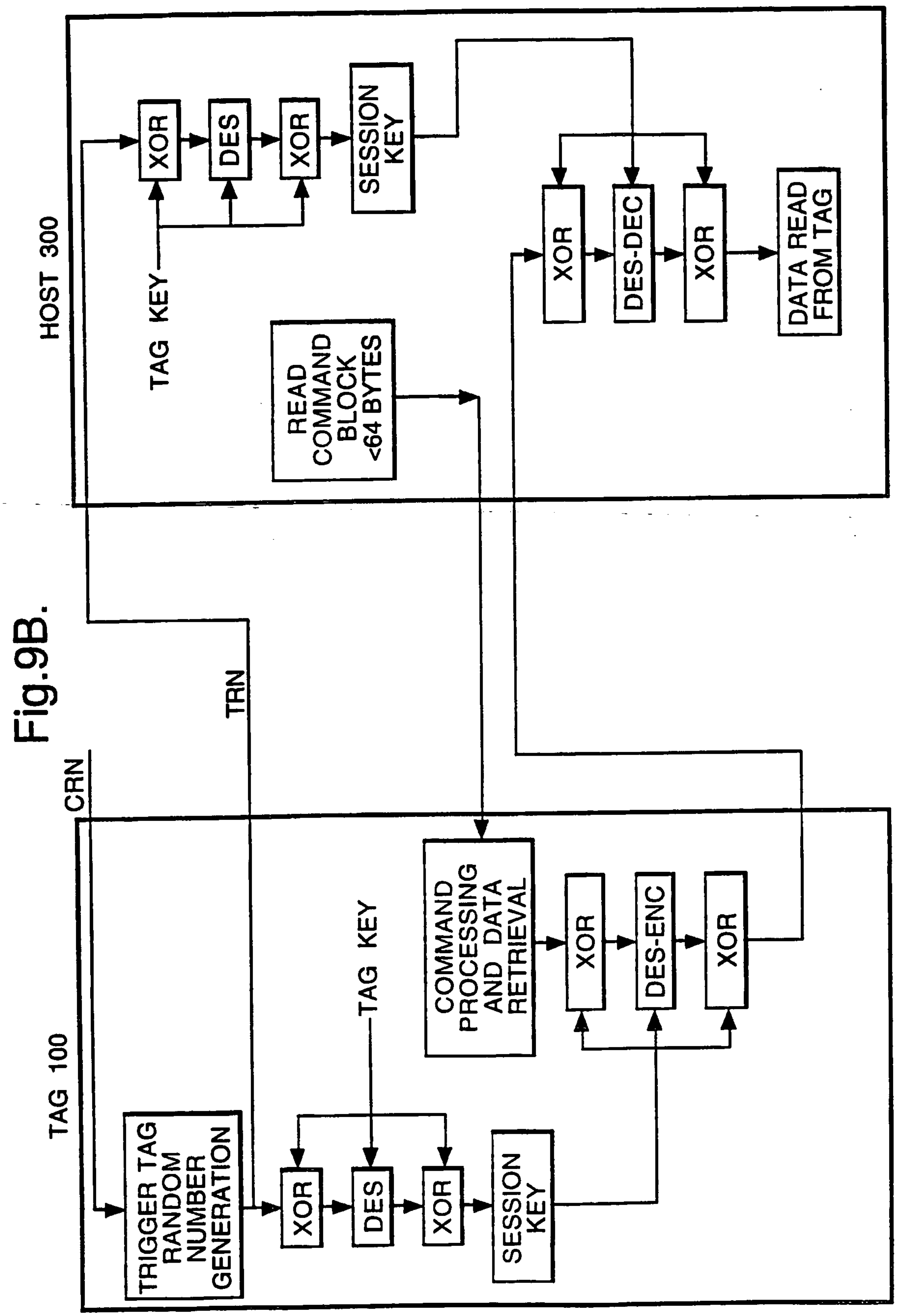

HOST 300
TAG KEY
XOR
DES
XOR
SESSION KEY
READ COMMAND BLOCK <64 BYTES
XOR
DES-DEC
XOR
DATA READ FROM TAG
TRN
CRN
TAG 100
TRIGGER TAG RANDOM NUMBER GENERATION
TAG KEY
XOR
DES
XOR
SESSION KEY
COMMAND PROCESSING AND DATA RETRIEVAL
XOR
DES-ENC
XOR

Fig.9B.

Fig.11.

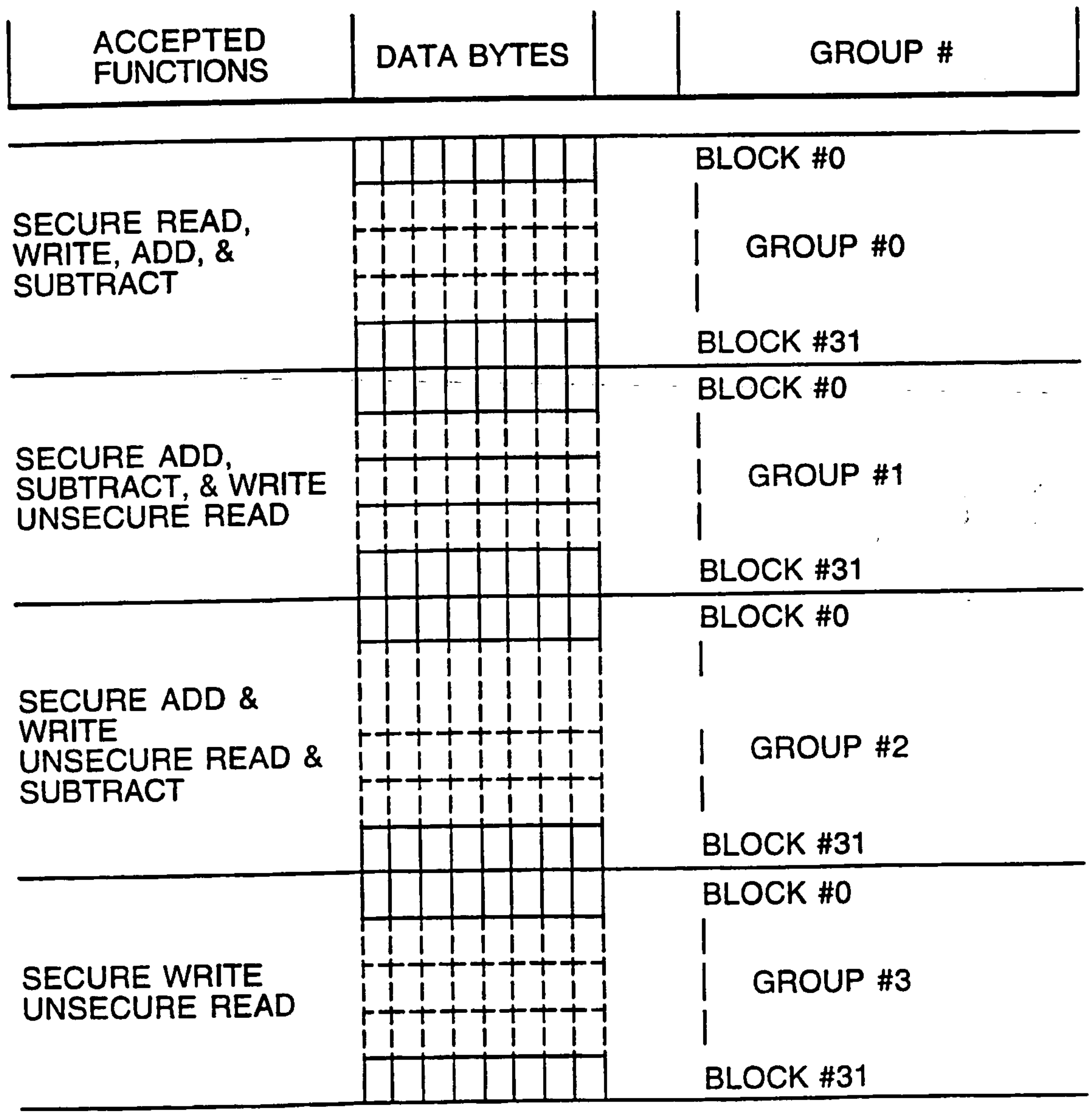

ACCEPTED FUNCTIONS
DATA BYTES
GROUP #
SECURE READ, WRITE, ADD, & SUBTRACT
BLOCK #0
GROUP #0
BLOCK #31
SECURE ADD, SUBTRACT, & WRITE UNSECURE READ
BLOCK #0
GROUP #1
BLOCK #31
SECURE ADD & WRITE UNSECURE READ & SUBTRACT
BLOCK #0
GROUP #2
BLOCK #31
SECURE WRITE UNSECURE READ
BLOCK #0
GROUP #3
BLOCK #31

Fig.12.

| PARTITION # | SIZE | PASSWORD | CONTENTS | LOCAL ACCESS | HOST ACCESS |
|---|---|---|---|---|---|
| 0 | 192 BYTES | SUPERUSER KNOWN ONLY BY THE HOST. DIFFERENT PASSWORD FOR EACH TAG, CALCULATED FROM THE TAG ID # THE MSB OF THE PASSWORD IS FORCED TO ZERO | CUSTOMER DEFINITION DATA FIELDS | READ ANY FIELDS WITHOUT A PASSWORD. WRITE ONLY ON SPECIFIC INSTRUCT-IONS WITH THE PASSWORD PASSED FROM THE HOST | AS NOTED AT LEFT |
| 1 | 32 BYTES | ADMINISTRA-TOR PASSWORD. CALCULATED FROM THE TAG ID NUMBER. KNOWN BY THE CRIND AND HOST. THE MSB OF THE PASSWORD IS FORCED TO A LOGICAL ONE | LOCAL AND HOST AUTHENT-ICATION NUMBERS. CRIND CAN CALCULATE THE LOCAL AUTH# ONLY THE HOST CAN CALCULATE THE HOST AUTH#. | CRIND CAN CALCULATE THE PASSWORD FOR THIS PARTITION FROM THE TAG ID. CAN THEN READ LOCAL AND HOST AUTHENT ICATION CODES. | THE HOST AUTHENT-ICATION NUMBER IS PASSED TO THE HOST FOR AUTHENT-ICATION |
| 2 | 32 BYTES | USER PASSWORD. SAME PASSWORD IN ALL TAGS, KNOWN TO THE CRINDS. THE MSB OF THE PASSWORD IS FORCED TO ZERO | SCRATCH-PAD READ/WRITE AREA | COMPLETE READ/WRITE ACCESS LOCALLY WITH A KNOWN PASSWORD | N/A |

Fig.13A.

| FIELD NUMBER | FIELD SIZE (BYTES) | FIELD FORMAT | DESCRIPTION | INITIAL VALUES IN HEX |
|---|---|---|---|---|
| 1 | 1 | BINARY | NUMBER OF VALID BYTES IN PARTITION 0 | 14 |
| 2 | 1 | BINARY | TAG DATA FORMAT SPECIFIER | 00 |
| 3 | 1 | BINARY | 0-NETWORK TRANSPONDER<br>1-LOCAL USE TRANSPONDER | 00 |
| 4 | 1 | BINARY | NUMBER TO DESIGNATE THE TYPE OF TRANSPONDER<br>00-NOT USED<br>01-PASSIVE KEY CHAIN<br>02-PASSIVE CARD<br>03-PASSIVE CAR MOUNT<br>04-128-PASSIVE OTHER<br>128-ACTIVE KEY CHAIN<br>129-ACTIVE CARD<br>130-ACTIVE CAR MOUNT<br>131-ACTIVE KEY CHAIN WITH DES<br>132-ACTIVE CARD WITH DES<br>133-ACTIVE CAR MOUNT WITH DES<br>134-255-ACTIVE OTHER | 80 |
| 5-10 | 6 | NUMERIC | ISSUE DATE MM/YYYY | 30 35 31 39 39 37 |
| 11 | 1 | BINARY | NUMBER TO INDICATE THE ACCEPTABLE FUEL GRADE<br>0-NO RESTRICTION | 00 |
| 12 | 1 | BINARY | PURCHASE RESTRICTIONS<br>0-NO RESTRICTIONS<br>1-FUEL ONLY<br>2-FUEL AND OIL ONLY<br>3-DRIVER ID TAG, NO PURCHASE ALLOWED | 00 |

Fig.13B.

| | | | | |
|---|---|---|---|---|
| 13-16 | 4 | NUMERIC | FLEET CARD REQUIREMENTS ALLOWS UP TO FOUR PROMPTS TO BE SPECIFIED IN THIS FIELD<br>0000-NOT A FLEET CARD<br>1-PROMPT FOR ENTER DRIVER NUMBER<br>2-PROMPT FOR ENTER VEHICLE NUMBER<br>3-PROMPT FOR ENTER JOB NUMBER<br>4-PROMPT FOR ENTER DATA<br>5-PROMPT FOR ENTER DEPARTMENT NUMBER<br>6-PROMPT FOR ENTER USER ID<br>7-RESERVED FOR FUTURE EXPANSION<br>8-PROMPT FOR ENTER ODOMETER<br>9-RESERVED FOR FUTURE EXPANSION<br>A-REQUIRE A DRIVER ID TAG | 30 30 30 30 |
| 17 | 1 | BINARY | PRINT RECEIPT<br>0-NO RECEIPT (OR HOST DEFINED)<br>1-PRINT RECEIPT AFTER FUELING<br>2-255-TBD | 00 |
| 18 | 1 | BINARY | REQUIRE PIN NUMBER ENTRY<br>0-NO PIN NUMBER REQUIRED (OR HOST DEFINED)<br>1-PIN NUMBER ENTRY REQUIRED<br>2-255-TBD | 00 |
| 19 | 1 | BINARY | PAYMENT TYPE<br>0-NETWORK DEFINED<br>1-255-OTHER TBD | 00 |
| 20 | 1 | BINARY | DISPENSER AUTHORISATION<br>0-WAIT FOR NETWORK<br>1-AS SOON AS CRIND ALLOWS<br>2-255-TBD | 00 |
| 21-192 | 172 | UNKNOWN | UNDESIGNATED BINARY FIELDS, CODED AS REQUIRED | ALL 00 |

Fig.14.

| FIELD NUMBER | FIELD SIZE (BYTES) | FIELD FORMAT | DESCRIPTION | INITIAL VALUES IN HEX |
|---|---|---|---|---|
| 193-200 | 8 | BINARY | LOCAL AUTHENTICATION CODE | CALCULATED AT HOST |
| 201-208 | 8 | BINARY | HOST AUTHENTICATION CODE | CALCULATED AT HOST |
| 209-216 | 8 | BINARY | UNUSED | ALL 00 |
| 217-224 | 8 | BINARY | UNUSED | ALL 00 |

Fig.15.

| FIELD NUMBER | FIELD SIZE (BYTES) | FIELD FORMAT | DESCRIPTION | INITIAL VALUES IN HEX |
|---|---|---|---|---|
| 225-256 | 32 | BINARY | LOCAL SCRATCH PAD AREA | ALL 00 |

Fig.16.

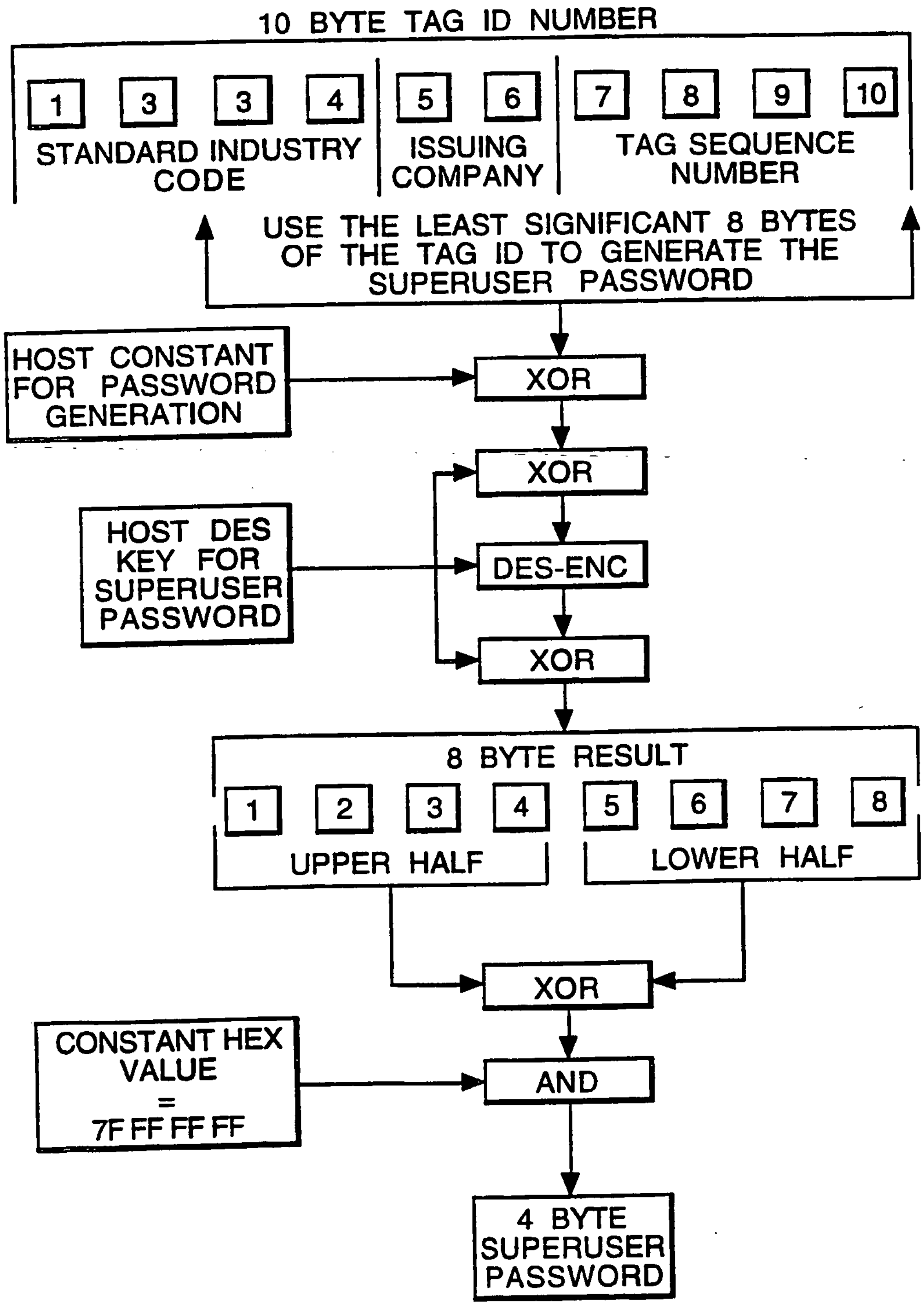

10 BYTE TAG ID NUMBER
1
3
3
4
5
6
7
8
9
10
STANDARD INDUSTRY CODE
ISSUING COMPANY
TAG SEQUENCE NUMBER
USE THE LEAST SIGNIFICANT 8 BYTES OF THE TAG ID TO GENERATE THE SUPERUSER PASSWORD
HOST CONSTANT FOR PASSWORD GENERATION
XOR
XOR
HOST DES KEY FOR SUPERUSER PASSWORD
DES-ENC
XOR
8 BYTE RESULT
1
2
3
4
5
6
7
8
UPPER HALF
LOWER HALF
XOR
CONSTANT HEX VALUE = 7F FF FF FF
AND
4 BYTE SUPERUSER PASSWORD

Fig.17.

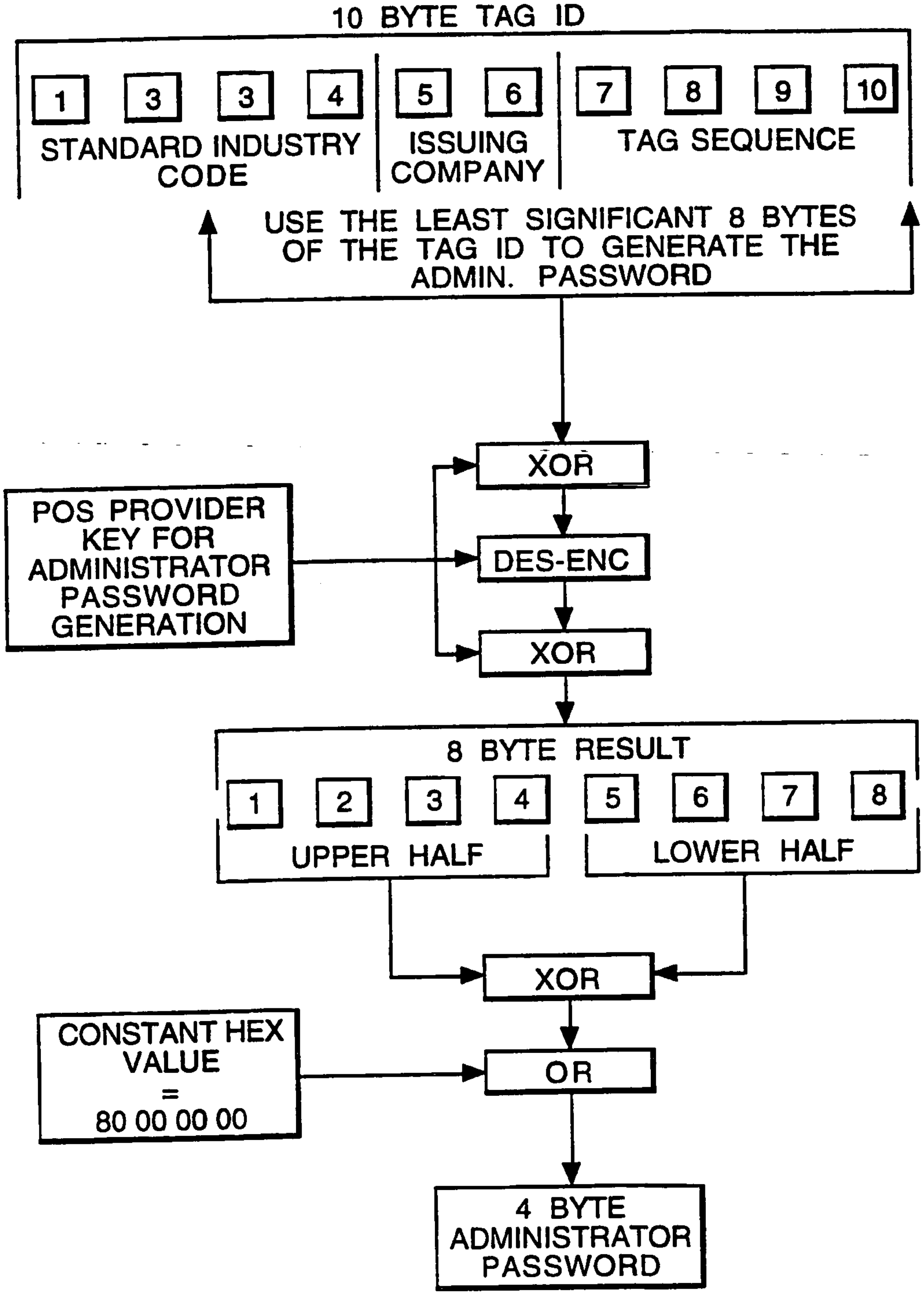

10 BYTE TAG ID
1
3
3
4
5
6
7
8
9
10
STANDARD INDUSTRY CODE
ISSUING COMPANY
TAG SEQUENCE
USE THE LEAST SIGNIFICANT 8 BYTES OF THE TAG ID TO GENERATE THE ADMIN. PASSWORD
XOR
POS PROVIDER KEY FOR ADMINISTRATOR PASSWORD GENERATION
DES-ENC
XOR
8 BYTE RESULT
1
2
3
4
5
6
7
8
UPPER HALF
LOWER HALF
XOR
CONSTANT HEX VALUE = 80 00 00 00
OR
4 BYTE ADMINISTRATOR PASSWORD

Fig.18.

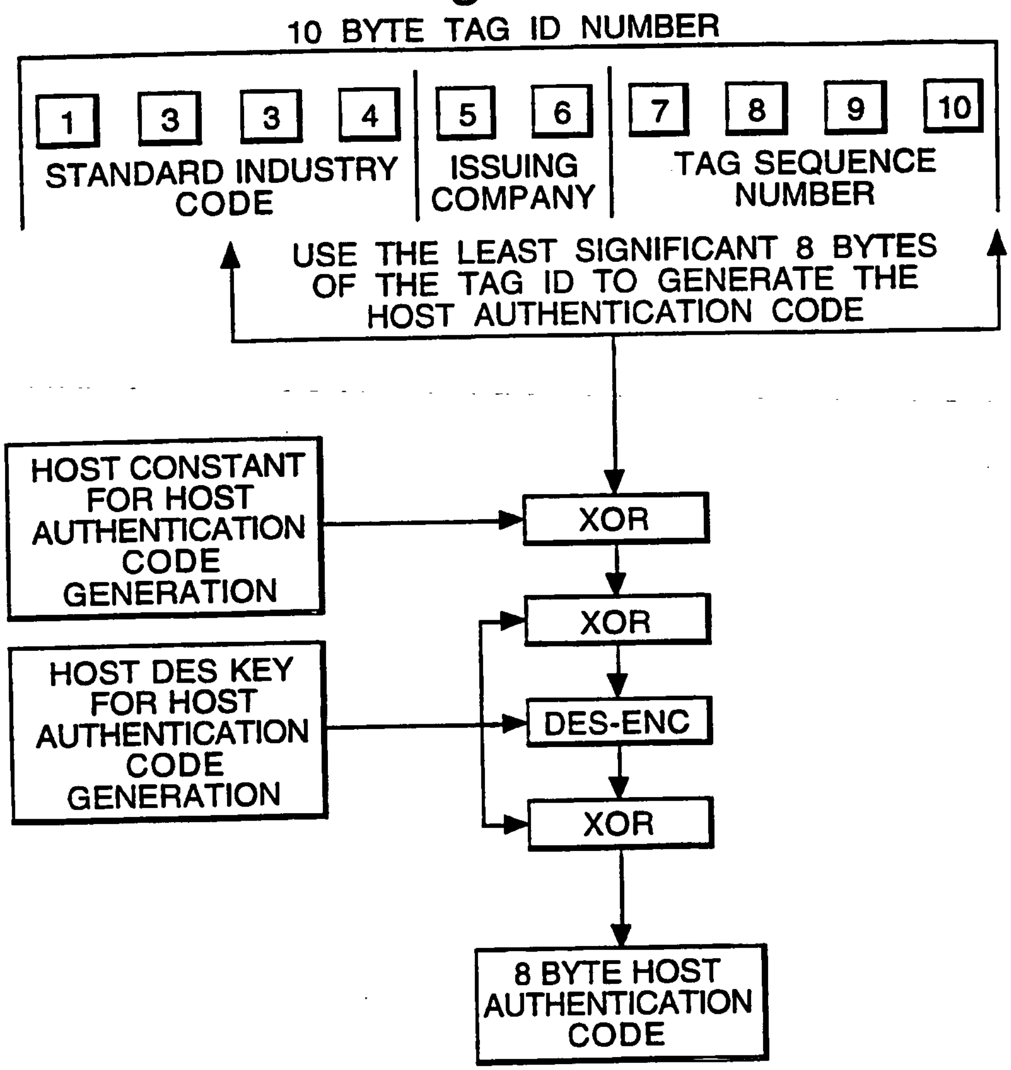
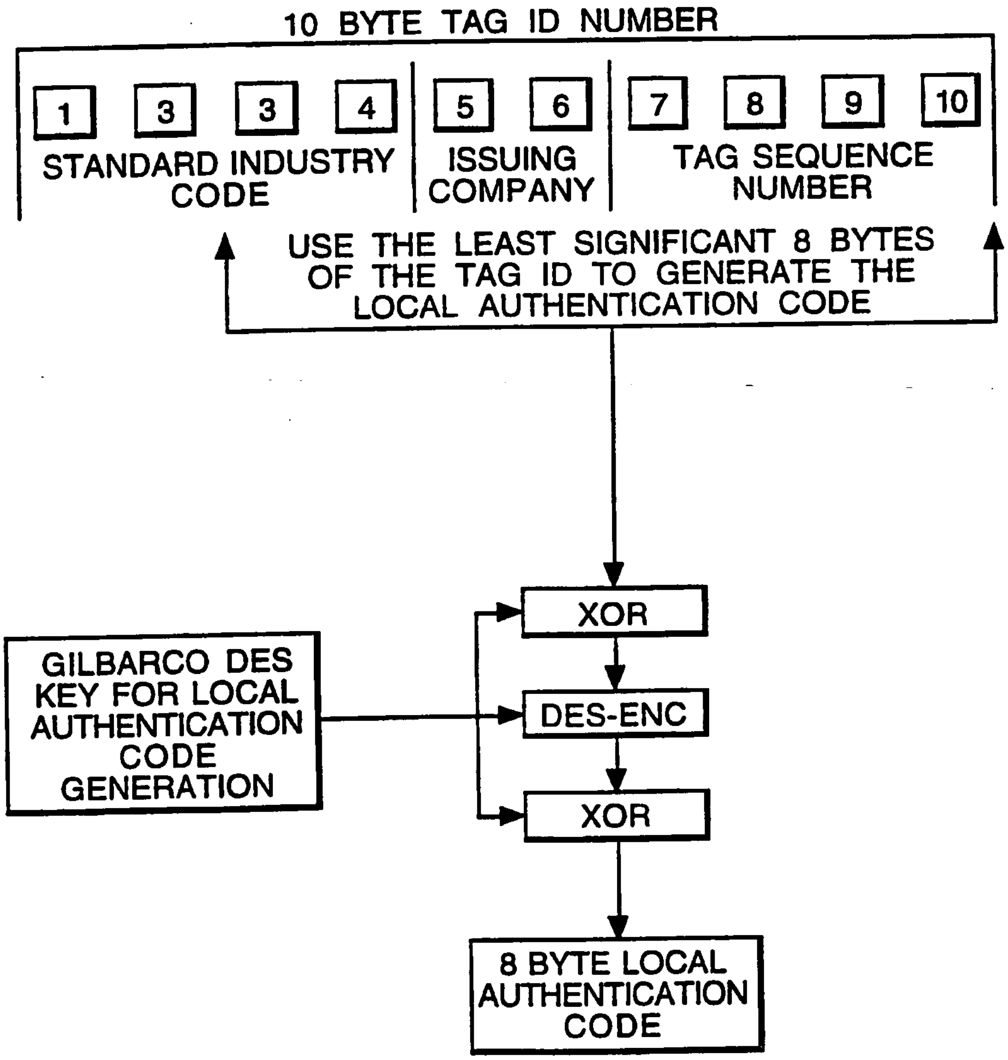

10 BYTE TAG ID NUMBER
1 3 3 4
STANDARD INDUSTRY CODE
5 6
ISSUING COMPANY
7 8 9 10
TAG SEQUENCE NUMBER
USE THE LEAST SIGNIFICANT 8 BYTES OF THE TAG ID TO GENERATE THE HOST AUTHENTICATION CODE
HOST CONSTANT FOR HOST AUTHENTICATION CODE GENERATION
XOR
XOR
HOST DES KEY FOR HOST AUTHENTICATION CODE GENERATION
DES-ENC
XOR
8 BYTE HOST AUTHENTICATION CODE

Fig.19.


10 BYTE TAG ID NUMBER
1
3
3
4
5
6
7
8
9
10
STANDARD INDUSTRY CODE
ISSUING COMPANY
TAG SEQUENCE NUMBER
USE THE LEAST SIGNIFICANT 8 BYTES OF THE TAG ID TO GENERATE THE LOCAL AUTHENTICATION CODE
GILBARCO DES KEY FOR LOCAL AUTHENTICATION CODE GENERATION
XOR
DES-ENC
XOR
8 BYTE LOCAL AUTHENTICATION CODE

# INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 98/02083

**A. CLASSIFICATION OF SUBJECT MATTER**
IPC 6 G07F7/10 G07F13/02

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G07F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category ° | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| Y | EP 0 134 109 A (OKI ELECTRIC IND CO LTD) 13 March 1985 | 1,23,24, 30,31, 35-37, 41-44 |
| A | see claim 1; figure 3 | 2-22, 25-29, 32-34, 38-40 |
| | --- | |
| Y | WO 97 24689 A (DRESSER IND ;MOBIL OIL CORP (US)) 10 July 1997 | 1,23,24, 30,31, 35-37, 41-44 |
| A | see claim 1; figure 1 | 2-22, 25-29, 32-34, 38-40 |
| | --- | |
| | -/-- | |

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 16 October 1998 | 26/10/1998 |

| Name and mailing address of the ISA | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,<br>Fax: (+31-70) 340-3016 | Kirsten, K |

1

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

| Category ° | Citation of document, with indication,where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | US 5 578 808 A (TAYLOR DOUGLAS C) 26 November 1996 see claim 1; figures 1,4 | 1-44 |
| A | US 5 521 363 A (TANNENBAUM DAVID H) 28 May 1996 see claim 1; figure 5 | 1-44 |
| A | EP 0 758 777 A (PALOMAR TECHN CORP) 19 February 1997 see claim 1; figure 1 | 1-44 |
| A | WO 96 36947 A (NAT WESTMINSTER BANK PLC ;EVERETT DAVID BARRINGTON (GB); RICHARDS) 21 November 1996 see claim 1; figure 5 | 1-44 |

1

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/GB 98/02083

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| EP 0134109 A | 13-03-1985 | JP 1961424 C | 25-08-1995 |
| | | JP 6058670 B | 03-08-1994 |
| | | JP 60031672 A | 18-02-1985 |
| | | US 4590365 A | 20-05-1986 |
| WO 9724689 A | 10-07-1997 | AU 1432797 A | 28-07-1997 |
| US 5578808 A | 26-11-1996 | US 5530232 A | 25-06-1996 |
| | | AU 4573596 A | 05-09-1996 |
| | | CA 2170327 A | 29-08-1996 |
| | | CN 1138723 A | 25-12-1996 |
| | | DE 19607363 A | 19-09-1996 |
| | | GB 2298505 A | 04-09-1996 |
| US 5521363 A | 28-05-1996 | NONE | |
| EP 0758777 A | 19-02-1997 | CA 2182464 A | 11-02-1997 |
| WO 9636947 A | 21-11-1996 | AU 696468 B | 10-09-1998 |
| | | AU 5698596 A | 29-11-1996 |
| | | BG 102028 A | 29-05-1998 |
| | | CA 2220070 A | 21-11-1996 |
| | | CZ 9703530 A | 17-06-1998 |
| | | EP 0829070 A | 18-03-1998 |
| | | GB 2314663 A,B | 07-01-1998 |
| | | NO 975199 A | 12-11-1997 |
| | | PL 323313 A | 16-03-1998 |